Multimedia 7.6

User's Guide

The information contained herein is proprietary and confidential and cannot be disclosed or duplicated without the prior written consent of Genesys Telecommunications Laboratories, Inc.

Copyright © 2002–2009 Genesys Telecommunications Laboratories, Inc. All rights reserved.

About Genesys

Genesys Telecommunications Laboratories, Inc., a subsidiary of Alcatel-Lucent, is 100% focused on software for call centers. Genesys recognizes that better interactions drive better business and build company reputations. Customer service solutions from Genesys deliver on this promise for Global 2000 enterprises, government organizations, and telecommunications service providers across 80 countries, directing more than 100 million customer interactions every day. Sophisticated routing and reporting across voice, e-mail, and Web channels ensure that customers are quickly connected to the best available resource—the first time. Genesys offers solutions for customer service, help desks, order desks, collections, outbound telesales and service, and workforce management. Visit www.genesyslab.com for more information.

Each product has its own documentation for online viewing at the Genesys Technical Support website or on the Documentation Library DVD, which is available from Genesys upon request. For more information, contact your sales representative.

Notice

Although reasonable effort is made to ensure that the information in this document is complete and accurate at the time of release, Genesys Telecommunications Laboratories, Inc., cannot assume responsibility for any existing errors. Changes and/or corrections to the information contained in this document may be incorporated in future versions.

Your Responsibility for Your System's Security

You are responsible for the security of your system. Product administration to prevent unauthorized use is your responsibility. Your system administrator should read all documents provided with this product to fully understand the features available that reduce your risk of incurring charges for unlicensed use of Genesys products.

Trademarks

Genesys, the Genesys logo, and T-Server are registered trademarks of Genesys Telecommunications Laboratories, Inc. All other trademarks and trade names referred to in this document are the property of other companies. The Crystal monospace font is used by permission of Software Renovation Corporation, www.SoftwareRenovation.com.

Technical Support from VARs

If you have purchased support from a value-added reseller (VAR), please contact the VAR for technical support.

Technical Support from Genesys

If you have purchased support directly from Genesys, please contact Genesys Technical Support at the following regional numbers:

Region	Telephone	E-Mail
North and Latin America	+888-369-5555 or +506-674-6767	support@genesyslab.com
Europe, Middle East, and Africa	+44-(0)-1276-45-7002	support@genesyslab.co.uk
Asia Pacific	+61-7-3368-6868	support@genesyslab.com.au
Japan	+81-3-6361-8950	support@genesyslab.co.jp

Prior to contacting technical support, please refer to the *Genesys Technical Support Guide* for complete contact information and procedures.

Ordering and Licensing Information

Complete information on ordering and licensing Genesys products can be found in the ***Genesys 7 Licensing Guide*.**

Released by

Genesys Telecommunications Laboratories, Inc. www.genesyslab.com

Document Version: 76mm_us_04-2009_v7.6.101.02

Table of Contents

List of Procedures **7**

Preface **9**

Multimedia and the CIM Platform 10
- CIM Platform 10
- Multimedia 10

Intended Audience 12
Chapter Summaries 12
Document Conventions 13
Related Resources 14
Making Comments on This Document 16
Document Change History 17
- New in Version 7.6.1 17

Chapter 1 **Genesys Knowledge Management: Basics** **19**

Overview 19
- Functionality 19
- Components 20
- User Interface 21
- Introduction to Category Structure 22
- Using Basic Knowledge Manager Objects 24

Using Categories and Standard Responses 25
- Creating a Category Tree 25
- Creating Standard Responses 28
- Searching for Standard Responses 40
- Standard Responses for SMS Gateways 42

Field Codes 44
- Overview 44
- Using UCS Data in Standard Responses: System Variables 45
- Custom Variables 46
- Using Your Own Data in Standard Responses 46
- Field Codes in Knowledge Manager 47
- Field Code Anatomy 52
- Examples 55

Screening Rules 58
Screening Rule Editor 58
What Screening Rules Check 69
Syntax and Semantics 70
Examples 75
Importing and Exporting 77
Typical Response Times 82

Chapter 2 **Genesys Knowledge Management: Content Analyzer 83**

Overview 83
Training 84
Creating a Training Object 84
Copying a Training Object 88
Adding More Text Objects to the Training Object 91
Adding Uncategorized E-Mails Using the TO Data Analyzer 95
Schedule Training 108
Notes on Language 115
Large Training Objects 118
Testing Models 119
Cross-Validation 119
Testing a Model on a Training Object 120
Testing a Model on Composed Text 123
Testing a Model on Uncategorized Messages 125
Using and Rating Models 128
Reading and Understanding the Ratings 128
Reporting on the Ratings 135
Improving the Results 137
Design and Use Considerations 138
Design 138
When to Train 139
Applying the Ratings 140
Language Detection Model 141
FAQ Objects 142
Sample FAQ .jar File 143
More About FAQ Objects 145
Creating an FAQ Object 146
Full Category Tree Subtab: Configuring the Category Tree 148
FAQ Category Tree Subtab: Viewing and Testing 152
Generating an FAQ.jar File 155
Typical Response Times 156

Chapter 3 **Multi-Tenancy 159**

Overview 159
Configuration 159
Interaction Server 160
Limitations 160

Chapter 4 **Load Balancing and Backup Configuration 161**

Overview 161
Web API Server 162
Load-Balancing Configuration for Web API Server 163
Load-Balancing API 165
Interaction Server 165
Balancing Universal Routing Servers 166
Balancing Multimedia Application Servers 166
Balancing DB Servers 167
Backup Configuration 167
High-Availability Support 168

Chapter 5 **Ongoing Administration and Other Topics 169**

Administration 169
General Recommendations 169
UCS 171
Interaction Server 174
Classification Server 174
Web Compound Samples 175
Security 175
Limitations 176
General 176
Knowledge Manager 176
Chat Server 178
Attached Data 180
Interaction Server 180
E-mail Server Java 181
Co-Browsing Server 181
Unicode Character Support 181
E-mail Server Java: Advanced Topics 182
Handling Unparsable E-Mails 182
JavaMail Properties 182
Delivery Status Notification and Message Disposition Notification ... 185
Interaction Server: Advanced Topics 188
Custom Database Fields 188
Event Logger 190

Chapter 6 **Contact Identification and Creation 197**

Overview 197
Contact Identification 197
Default Behavior 198
Customizing 199
Contact Creation 203
Default Behavior 203
Customizing 203

Chapter 7 **Interaction Properties 205**

Overview 205
Interaction Properties 207
System Properties 207
Business Properties 210
Custom Properties 211
Other Properties 213
Changes in Release 7.6.1 213
Business Attributes 214
Translations 216

Chapter 8 **Transitioning to Multimedia from ICS 6.x 223**

Overview 223
Converting the Contact Server Database 224
Conversion Procedures 225
Corruption Recovery 231
Errors and Recovery 232
After Transition 233
Transition Results 233
Content Analysis Rules: Match Criteria (Multimedia Screening Rules) .. 233
Standard Responses, Categories, Feedback 234
Other 6.5.x Objects 238
Contact Center Objects 238
Content Analysis Rules 238
Strategies 238
Other Objects 239

Glossary **241**

Index **249**

List of Procedures

Creating a category tree 25
Creating or editing a standard response 28
Saving an existing user dictionary 31
Filling out the HTML part tab 32
Filling out the Additional tab 35
Searching for a standard response 40
Creating field codes 48
Inserting field codes into a standard response 50
Using a custom variable 55
Creating a rule 59
Testing screening rules on the UCS database 64
Testing screening rules on specially-created text 66
Finding a rule 68
Exporting Knowledge Manager objects 78
Importing Knowledge Manager objects 80
Creating a new training object 85
Making an exact copy of a training object 89
Making a filtered copy of a training object 89
Creating new e-mails manually 91
Adding standard responses or other training objects 93
Adding e-mails to a category on the Main tab 95
Example of use 107
Scheduling training using the Model Options tab 109
Creating and testing filters 112
Adapting the lexical analyzer 117
Creating two new training objects 120
Extracting random text objects 121
Testing a model on a training object 122
Producing an HTML report on ratings 136
Importing a language classification model 141

Adding more languages to the model 142
Creating a new FAQ object 146
Generating and testing an FAQ.jar file 155
Configuring an Application Cluster 164
Configuring a custom interaction property 189
Converting a field to a BLOB 190
Deploying Event Logger 191
Customize which attributes are checked 200
Customize the priority of attributes 200
Customize per media type 201
Turning contact creation on and off 203
Setting minimum attributes for creation 203
Using custom properties 212
Preliminaries to database transition 225
Preparing the database 228
Running the Transition Tool 229

Preface

Welcome to the Multimedia 7.6 User's Guide. This guide provides instructions on using Multimedia 7.6 features and functions.

This guide is valid only for the 7.6.x releases of this product.

Note: For releases of this document created for other releases of this product, please visit the Genesys Technical Support website, or request the Documentation Library DVD, which you can order by e-mail from Genesys Order Management at `orderman@genesyslab.com`.

This chapter provides an overview of this document, identifies the primary audience, introduces document conventions, and lists related reference information:

- Multimedia and the CIM Platform, page 10
- Intended Audience, page 12
- Chapter Summaries, page 12
- Document Conventions, page 13
- Related Resources, page 14
- Making Comments on This Document, page 16
- Document Change History, page 17

Multimedia and the CIM Platform

Genesys Multimedia (formerly Multi-Channel Routing) is a cover term for Genesys components that work together to manage interactions whose media is something other than traditional telephonic voice (for example, e-mail or chat).

Multimedia includes some parts of the Genesys Customer Interaction Management (CIM) Platform, plus certain of the media channels that run on top of the Platform.

CIM Platform

The CIM Platform consists of the following:

- Management Framework
- Reporting (CC Analyzer, CCPulse+)
- Interaction Management, which in turn consists of:
  - Universal Routing
  - Interaction Workflow
  - Knowledge Management
  - Content Analysis
  - Universal Contact History

On top of the CIM Platform are various media channels. Some, such as Genesys Network Voice, handle traditional telephony. Others, such as Genesys E-mail, handle other media.

Multimedia

Multimedia, then, consists of the following:

- From the CIM Platform, all of Interaction Management except for Universal Routing:
  - Interaction Workflow—centralized handling of interactions irrespective of media type
  - Knowledge Management—creation and maintenance of standard responses and screening rules
  - Content Analysis—optional enhancement to Knowledge Management, applying natural language processing technology to categorize interactions
  - Universal Contact History—storage of data on contacts and on interactions (linked as threads)

  Universal Routing is not considered part of Multimedia because it deals with both traditional telephonic interactions and the nontraditional interactions that are handled in Multimedia.

- From the media channels, at least one of the following:
  - Genesys E-mail—e-mail
  - Genesys Web Media—chat
  - Genesys Open Media—ability to add customized support for other media (fax, for example)
- Optionally, Web Collaboration—the ability for agents and customers to co-browse (simultaneously navigate) shared web pages. This is an option that you can add to either Genesys Web Media or Inbound Voice.

See Figure 1.

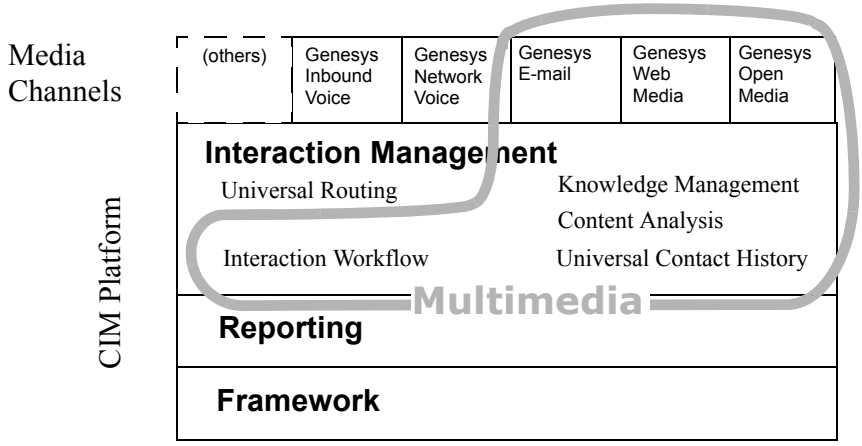


Figure 1: Multimedia in Relation to the CIM Platform and Media Channels

Note: Although Universal Routing is not considered part of Multimedia, any functioning solution (platform plus channels) that includes any part of the Interaction Management sector requires Universal Routing.

Licensing

Licensing requirements are:

- For each agent: one Multimedia Agent seat.
- For each media option: one media channel (E-mail and/or Web Media and/or custom media).
- For Genesys Content Analyzer: NLP Content Analysis license.

See also the *Genesys 7 Licensing Guide.*

Reporting

Reporting templates are available for Multimedia. For details see the *Reporting Technical Reference Guide for the Genesys 7.x Release.*

Intended Audience

This guide, primarily intended for all users involved in setting up Genesys Multimedia, assumes that you have a basic understanding of:

- E-mail and web technology.
- Network design and operation.
- Your own network configurations.

You should also be familiar with:

- Genesys Framework architecture and functions.
- Computer-telephony integration (CTI) concepts, processes, terminology, and applications.

Chapter Summaries

In addition to this opening chapter, this guide contains these chapters and a glossary:

- Chapter 1, "Genesys Knowledge Management: Basics," on page 19, describes categories, standard responses, screening rules, and field codes.
- Chapter 2, "Genesys Knowledge Management: Content Analyzer," on page 83, describes the content analysis and training functionalities of the Genesys Content Analyzer option.
- Chapter 3, "Multi-Tenancy," on page 159, provides information on configuring multi-tenant environments.
- Chapter 4, "Load Balancing and Backup Configuration," on page 161, provides information on load balancing.
- Chapter 5, "Ongoing Administration and Other Topics," on page 169, describes matters such as monitoring of database size and security planning.
- Chapter 6, "Contact Identification and Creation," on page 197, describes how Universal Contact Server (UCS) identifies existing contacts and creates new contacts.
- Chapter 7, "Interaction Properties," on page 205, provides details on parameters of Multimedia that are accessible in IRD (Interaction Routing Designer) objects.
- Chapter 8, "Transitioning to Multimedia from ICS 6.x," on page 223, provides information on converting the ICS 6.x contact server database and other 6.x objects for use in Multimedia 7.x.
- The glossary, which begins on page 241, defines terms used in Genesys Multimedia in general.

Document Conventions

This document uses some stylistic and typographical conventions with which you might want to familiarize yourself.

Document Version Number

A version number appears at the bottom of the inside front cover of this document. Version numbers change as new information is added to this document. Here is a sample version number:

70fr_ref_09-2004_v7.1.000.000

You will need this number when you are talking with Genesys Technical Support about this product.

Type Styles

Italic

In this document, italic is used for emphasis, for documents' titles, for definitions of (or first references to) unfamiliar terms, and for mathematical variables.

Examples:

- Please consult the *Genesys Migration Guide* for more information.
- *A customary and usual practice* is one that is widely accepted and used within a particular industry or profession.
- Do *not* use this value for this option.
- The formula, $x + 1 = 7$ where x stands for . . .

Monospace Font

A monospace font, which looks like `teletype or typewriter text`, is used for all programming identifiers and GUI elements.

This convention includes the *names* of directories, files, folders, configuration objects, paths, scripts, dialog boxes, options, fields, text and list boxes, operational modes, all buttons (including radio buttons), check boxes, commands, tabs, CTI events, and error messages; the values of options; logical arguments and command syntax; and code samples.

Examples:

- Select the `Show variables on screen` check box.
- Click the `Summation` button.
- In the `Properties` dialog box, enter the value for the host server in your environment.
- In the `Operand` text box, enter your formula.
- Click `OK` to exit the `Properties` dialog box.

- The following table presents the complete set of error messages T-Server distributes in `EventError` events.
- If you select `true` for the `inbound-bsns-calls` option, all established inbound calls on a local agent are considered business calls.

Monospace is also used for any text that users must manually enter during a configuration or installation procedure, or on a command line:

Example:

- Enter `exit` on the command line.

Information About Screen Captures Used in This Document

Screen captures taken from the product GUI (graphical user interface) and used in this document may sometimes contain a minor spelling, capitalization, or grammatical error. The text accompanying and explaining the screen captures corrects such errors *except* when such a correction would prevent you from installing, configuring, or successfully using the product. For example, if the name of an option contains a usage error, the name would be presented exactly as it appears in the product GUI; the error would not be corrected in any accompanying text.

Use of Square Brackets

In any logical arguments, commands, and programming syntax presented in this document, square brackets are used to indicate that a particular parametric value is optional. That is, the value is not required to resolve a command, argument, or programming syntax. The customer/user decides whether to supply a value and what that value is. Here is a sample:

```
smcp_server -host [/flags]s
```

Use of Angle Brackets

Angle brackets are used to indicate that a value in a logical argument, command, or programming syntax is required, but that the user must supply the data for the value. Because the value is specific to an individual enterprise—for example, DNs or port numbers—the program cannot predict (that is, program in) what the value is. Here is a sample:

```
smcp_server -host <confighost>
```

Related Resources

Consult these additional resources as necessary:

- *Multimedia 7.6 Deployment Guide,* which describes deployment procedures for all Multimedia components.

- *Multimedia 7.6 Reference Manual,* which provides a reference listing of all configuration options and of field codes used in standard responses.
- *Multimedia 7.6 Universal Contact Server Manager Help,* which is a guide to the Universal Contact Server Manager user interface.
- *Multimedia 7.6 Knowledge Manager Help,* which is a guide to the Knowledge Manager user interface.
- *Multimedia 7.6 Web API Reference,* which is a Javadoc listing of classes, methods, fields, and constants of the Web API portion of the Web API Server component.
- *Multimedia 7.6 Web API Client Developer's Guide,* which describes the structure of the Web API, explains the Simple and Compound Samples, and describes procedures for customizing them.
- *Genesys 7 Events and Models Reference Manual,* which includes a set of basic interaction models, showing the components involved and the event messages sent among them. These models and events were formerly presented in the *Open Media Interaction Models Reference Manual.* The request messages that were also described in that book are now documented in the API References of the Platform SDK.
- For the Web Collaboration option, the following documents describing design and administration for the KANA Response Live Server, which is supplied by Genesys as part of the Web Collaboration product:
  - *Hipbone Client API Reference Guide*
  - *KANA Response Live Organization Administration*
  - *KANA Response Live Server Installation Guide*
  - *KANA Response Live System Administration Tool User Guide*
- *Genesys Desktop 7.6 Deployment Guide,* which describes deployment procedures for the Genesys Desktop.
- *Genesys Desktop 7.6 Developer's Guide,* which describes customizing the Genesys Desktop.
- *Genesys Desktop 7.6 Agent Help,* which is a guide to the Genesys Agent Desktop.
- *Genesys Desktop 7.6 Supervisor's Help,* which is a guide to the Genesys Supervisor Desktop.
- *Universal Routing 7.6 Reference Manual,* which contains descriptions of all routing strategy objects, including those that are specific to Multimedia.
- *Universal Routing 7.5 Strategy Samples* (or later), which describes the sample strategies supplied with Universal Routing.
- *Universal Routing 7.5 Business Process User's Guide* (or later) which contains step-by-step instructions for using Interaction Routing Designer to design interaction workflows. It also describes the sample business processes supplied with Multimedia.

- *Universal Routing 7.6 Interaction Routing Designer Help,* which is a guide to Interaction Routing Designer, including the portion of it that designs interaction workflows and business processes for Multimedia.
- "Multimedia Log Events" in *Framework 7.6 Combined Log Events Help,* which is a comprehensive list and description of all events that may be recorded in logs.
- The *Genesys Technical Publications Glossary,* which ships on the Genesys Documentation Library DVD and which provides a comprehensive list of the Genesys and CTI terminology and acronyms used in this document.
- The *Genesys Migration Guide*, also on the Genesys Documentation Library DVD, which contains a documented migration strategy for Genesys product releases 5.x and later. Contact Genesys Technical Support for additional information.
- The Release Notes and Product Advisories for this product, which are available on the Genesys Technical Support website at http://genesyslab.com/support.
- The documentation on the other three members of the Genesys Customer Interaction Platform: Universal Routing, Reporting, and Management Framework.

Information on supported hardware and third-party software is available on the Genesys Technical Support website in the following documents:

- *Genesys Supported Operating Environment*
- *Genesys Supported Media Interfaces*

Genesys product documentation is available on the:

- Genesys Technical Support website at http://genesyslab.com/support.
- Genesys Documentation Library DVD, which you can order by e-mail from Genesys Order Management at orderman@genesyslab.com.

Making Comments on This Document

If you especially like or dislike anything about this document, please feel free to e-mail your comments to Techpubs.webadmin@genesyslab.com.

You can comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this document. Please limit your comments to the information in this document only and to the way in which the information is presented. Speak to Genesys Technical Support if you have suggestions about the product itself.

When you send us comments, you grant Genesys a nonexclusive right to use or distribute your comments in any way it believes appropriate, without incurring any obligation to you.

Document Change History

This section lists topics that are new in the current release of this document, or that have changed significantly from the preceding release.

New in Version 7.6.1

The following topics have been added or significantly changed since the initial 7.6 release:

- "Language and Dictionary Names" on page 30 in Chapter 1

  New section related to selecting language when checking spelling of standard responses.
- "The Attachments Tab" on page 37 in Chapter 1

  How to view an attachment to a standard response.
- "Searching for Standard Responses" on page 40 in Chapter 1

  Additional search terms available.
- "Screening Rule Editor" on page 58 in Chapter 1

  Step 6 describes screening by inbox rather than by To address.
- "Creating a Training Object" on page 84 in Chapter 2

  Step 4 explains using agent groups in training objects.
- "Interaction Server: Advanced Topics" on page 188 in Chapter 5

  New section describes custom database fields and Event Logger.
- "Changes in Release 7.6.1" in Chapter 7

  New section describes changes in the Interaction Server database.
- Table 24, "System Properties," on page 207 in Chapter 7

  New properties `PlaceInQueueSeq`, `ScheduledAt`, and `SubmitSeq` have been added.

Chapter

1 Genesys Knowledge Management: Basics

This chapter describes the use of the basic Knowledge Management functionalities. It covers these topics:

- Overview, page 19
- Using Categories and Standard Responses, page 25
- Field Codes, page 44
- Screening Rules, page 58
- Importing and Exporting, page 77
- Typical Response Times, page 82

Use this chapter in conjunction with *Multimedia 7.6 Knowledge Manager Help.*

The use of Genesys Content Analyzer, which is an optional enhancement to Knowledge Management, is described in Chapter 2, "Genesys Knowledge Management: Content Analyzer," on page 83.

Overview

This section provides an overview of Knowledge Management: its components, its functionalities, and how they are displayed in the user interface.

Functionality

Knowledge Management functionalities fall into the following four groups:

- **Categories/standard responses/field codes.** A system of categories, organized in a tree structure, provides the means of organizing *standard responses,* which are prewritten responses to interactions. Field codes

provide a way to particularize the standard response to individual interactions. Category trees are also integral to the classification functionality of Genesys Content Analyzer (see the third item in this list). You use Knowledge Manager to create category trees, and to create and edit the standard responses and the field codes that they can contain.

- **Screening rules.** Screening rules perform pattern matching on incoming interactions. The results of the pattern matching are then available for use in subsequent steps in routing and in interaction workflows. You use Knowledge Manager to create and edit the screening rules.
- **Genesys Content Analyzer.** This optional functionality uses natural language processing to analyze incoming interactions and assign them to categories in a category tree. Content analysis uses *models,* which are statistical representations of category trees. Models are produced by *training* on a collection of precategorized text objects (e-mails and other types). Knowledge Manager controls the training process and displays information about models. For details see Chapter 2, "Genesys Knowledge Management: Content Analyzer," on page 83.
- **FAQ.** With Content Analyzer, you can convert your category structure and standard responses into an FAQ list. You can either post the resulting FAQ list as text on your web site or use it as the source for an automatic question-answering facility.

Note: Many of the terms and concepts relevant to Genesys Knowledge Management are also defined in the glossary, which begins on page 241.

Components

Knowledge Management consists of the following components:

- **Classification Server** applies screening rules when triggered to do so by a routing strategy. In Genesys Content Analyzer, Classification Server also applies models to categorize incoming interactions. Both screening rules and models are stored in the Universal Contact Server database.
- **Training Server** trains the system to recognize categories. It is active only in the Genesys Content Analyzer.
- **Knowledge Manager** is the user interface.

Figure 2 shows the Knowledge Management components.

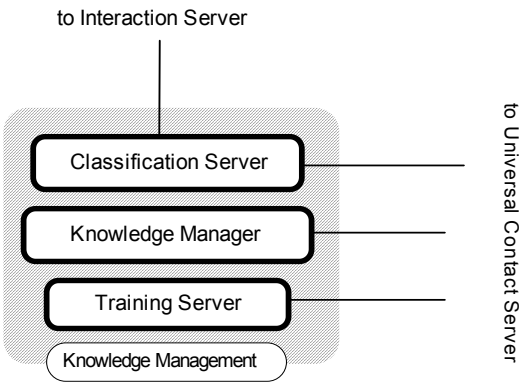


Figure 2: Knowledge Management and Other Relevant Components

User Interface

The Knowledge Manager user interface is a window with three (basic Knowledge Manager) or seven (Genesys Content Analyzer) tabs. Figure 3 shows the interface with seven tabs. It also shows the four panes Category Tree, Subcategories, Standard Responses, and Attached Rules.

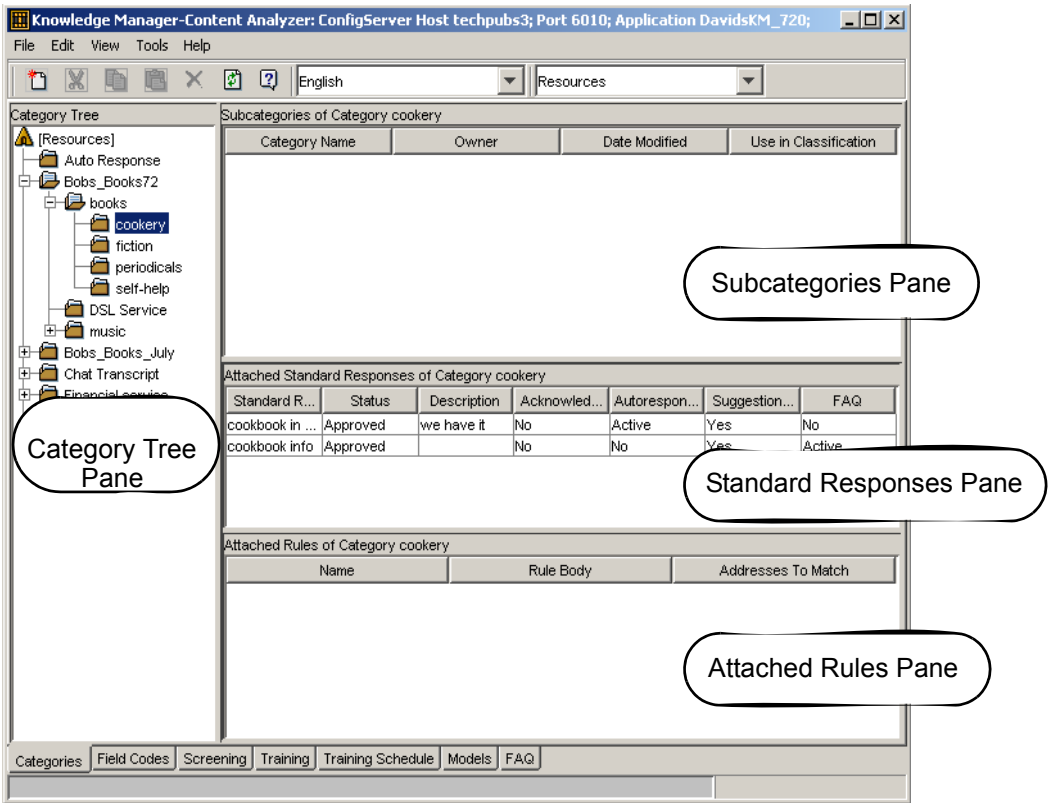


Figure 3: Knowledge Manager Window

Note: All Knowledge Manager screenshots in this guide show the Genesys Content Analyzer version. This differs from the version without Genesys Content Analyzer, even on tabs that are common to both. For example, without Genesys Content Analyzer, there is no `Tools` menu.

Table 1 briefly describes each tab and what it does. Note that the first three tabs are relevant to all users of Knowledge Manager, whereas the last four tabs are relevant only to users of Genesys Content Analyzer. For details see *Multimedia 7.6 Knowledge Manager Help*.

Table 1: Knowledge Manager Tabs

Relevance	Tab	Description
General	Categories	Displays and gives access to category trees and the standard responses and screening rules associated with categories. Provides access to the `Standard Response Editor`.
	Field Codes	Displays the set of field codes and provides access to the `Field Code Editor`
	Screening	Displays the set of screening rules and provides access to the `Screening Rules Editor`
Genesys Content Analyzer only	Training	Displays the set of training objects and provides access to the `Mail Editor`
	Training Schedule	Displays the start time, status, and other information about training objects
	Models	Displays information about models
	FAQ	Displays information about FAQ objects

Note: In most Knowledge Manager windows, lists of items can be sorted by clicking on the column headings. The major exceptions are lists of test results on the `Models` tab (see "Testing Models" on page 119 and "Using and Rating Models" on page 128).

Introduction to Category Structure

In general terms, a *category* is a unit of knowledge. Categories are organized in a tree structure; Figure 4 shows an example.

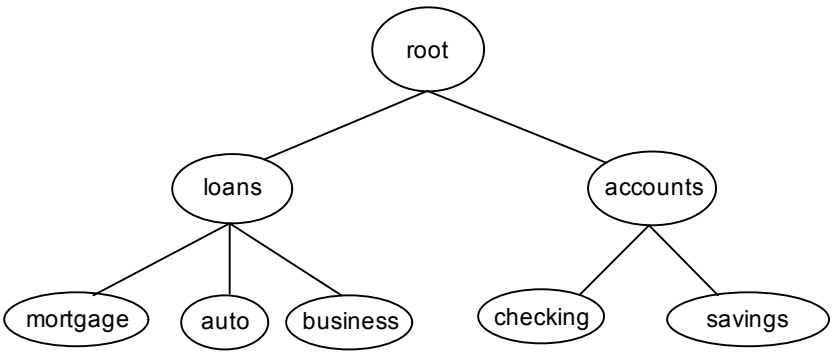


Figure 4: Example Category Tree

Genesys Multimedia uses category trees in the following ways:

- **Standard Responses.** The category tree is the means of organizing and providing access to the library of standard responses. Each standard response must be associated with one category. One category can have zero or many standard responses associated with it.
- **Classification and Routing.** Genesys Content Analyzer can classify an incoming e-mail in terms of the category tree. You can use the resulting classification in three ways:
  - To supply a standard response as an acknowledgment or an automatic response to an e-mail.
  - To suggest a standard response to the agent handling the e-mail.
  - To route the e-mail.

Categories with no associated standard responses may be of use in grouping other categories together.

Note these definitions:

- A *terminal* category is one that has no subcategories: a leaf on the category tree.
- A *nonterminal* category is one that has subcategories.
- *Child* is another term for subcategory. For example, in Figure 4, `savings` is a child of `accounts`, and `accounts` has the two children `checking` and `savings`.

A category tree is specific to a tenant and a language. Each tenant/language pair can have multiple category trees. Select tenant and language using the drop-down lists near the top of the Knowledge Manager window.

Without Content Analyzer, language is simply a label. You can design different sets of screening rules (for example) for different languages within a single tenant. But the screening rules operate the same way regardless of which language they are grouped under. With Content Analyzer, choice of language affects the way the system operates; for example, when classifying interactions. See “Notes on Language” on page 115 for more information on this topic.

Category membership is inherited. That is, if Category 1 includes Categories 10 and 11, and Category 10 includes Categories 100 and 101, then Category 1 also includes Categories 100 and 101.

Using Basic Knowledge Manager Objects

The basic Knowledge Manager objects are categories, standard responses, and screening rules. Table 2 presents an overall task flow for creating and managing these objects, and for importing and exporting them.

Table 2: Task Flow for Basic Knowledge Manager Objects

Objective	Related Procedures and Actions
1. Create a category tree.	See "Creating a category tree" on page 25.
2. Create and manage standard responses.	1. Open the Standard Response Editor and use the `General` tab to create a name and plain text content for a standard response: page 28. 2. Use the `HTML part` tab to create an HTML version of the standard response: page 32. 3. Use the `Additional` tab to specify possible uses and other attributes of the standard response: page 35. 4. Use the `Attachments` tab to add and manage attachments: page 37. 5. Use the `History` tab for version control: page 38. 6. Search for standard responses: page 40 7. Create standard responses for use with SMS gateways: page 42
3. Create field codes and use them in standard responses.	1. Create field codes: page 48. 2. Insert field codes in standard responses: page 50. 3. Study two examples of the use of field codes: page 55.
4. Create and manage screening rules.	1. Create screening rules: page 59. 2. Test screening rules: page 64 and page 66. 3. Search for screening rules: page 68.
5. Import and export Knowledge Manager objects.	1. Export: page 78. 2. Import: page 80.

Using Categories and Standard Responses

This section describes the creation and editing of categories and standard responses. It also provides some explanation of how they work.

Creating a Category Tree

You create categories on the `Categories` tab of Knowledge Manager.

Procedure: Creating a category tree

Purpose: To create a category tree for organizing and providing access to standard responses (and to additional functions in Genesys Content Analyzer).

Summary

Create categories on the `Categories` tab of Knowledge Manager.

Start of procedure

1. Decide whether you want a new root category or a subcategory of an existing category:
   a. For a root category, select the tenant, which is the topmost node in the category. Be sure that you also select the correct language.
   b. For a subcategory of an existing category, select the existing category on the Category Tree pane.
2. Select `New category` from the `File` menu (or right-click in the Category Tree pane or the Subcategories pane).

See Figure 5 on page 26, where the new category being created will be a subcategory of `books`.

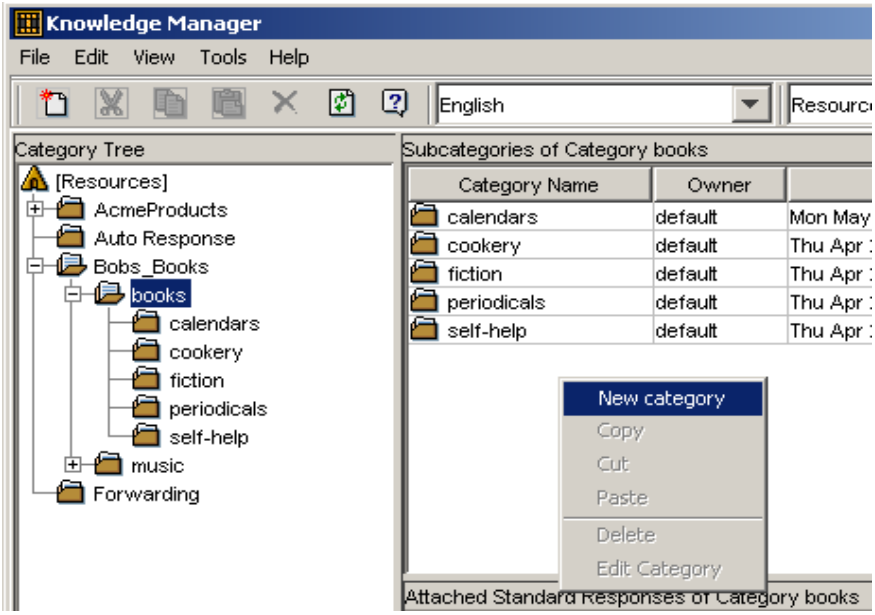


Figure 5: Creating a New Category

Note: Names of categories, like those of all Knowledge Manager objects, can consist only of alphanumeric characters (A–Z, a–z, 0–9), hyphen (-), underscore (_), and space. Names must also be no more than 64 characters long. For a reason to keep name length well below 64 characters, see page 81.

3. Selecting `New category` produces different dialog boxes, depending on whether your new category is a root or a subcategory.
   a. The `New root category` dialog box, shown in Figure 6, requires that you enter a name for the category:

Figure 6: New Root Category Dialog Box

   b. The `New category` dialog box, shown in Figure 7 on page 27, creates a subcategory. Enter a name for the category.

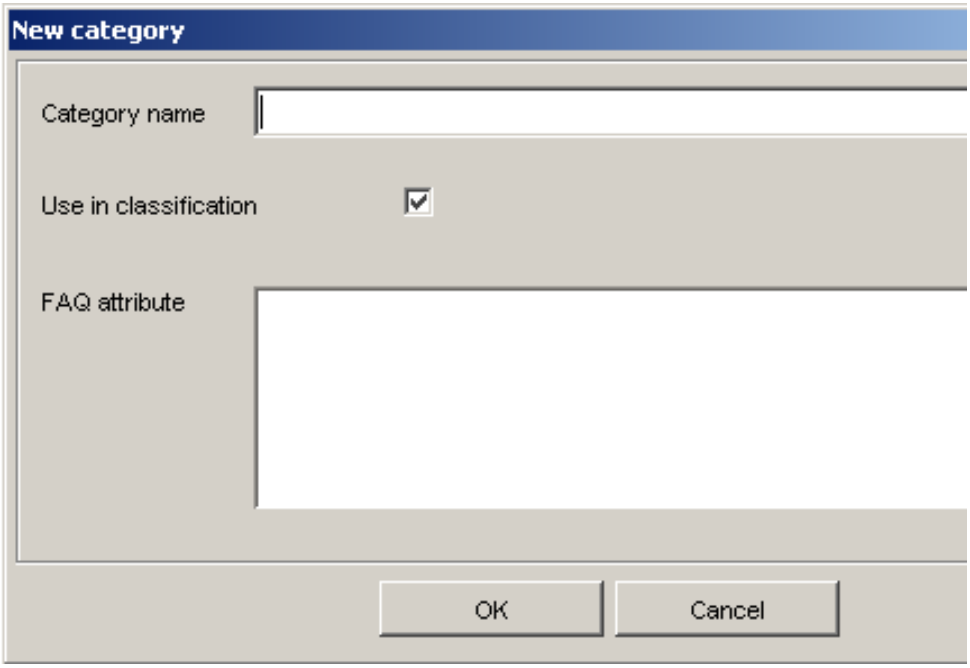


Figure 7: New Category Dialog Box

The `Use in Classification` check box enables you to choose whether Genesys Content Analyzer uses the category in classification. You may wish to have some categories that are used only for organizing other categories or standard responses, not for classification.

In the `FAQ attribute` box you can enter the text of a question to which a standard response associated with this category can serve as the answer. See "FAQ Objects" on page 142.

You can also edit an existing nonroot category by right-clicking it and selecting `Edit category`.

End of procedure

Next Steps

- Read further information about "Copying, Pasting, and Deleting" categories and category trees (this page), and about "Considerations in Designing a Category Tree" (page 28).
- Create standard responses (page 28).

Copying, Pasting, and Deleting

You can copy, paste, edit, and delete categories and category trees and their associated standard responses. Any of these operations on a category also applies to all of its subcategories.

If a category tree is used by a training object, you cannot delete the entire tree unless you first delete the training object. You can, however, delete any nonroot category from the tree (you cannot delete the root category). Training objects are described in "Training" on page 84.

Considerations in Designing a Category Tree

Without Genesys Content Analyzer, the sole function of categories is to organize the library of standard responses. With Genesys Content Analyzer, additional considerations come into play, as described in "Design and Use Considerations" on page 138.

Creating Standard Responses

You create standard responses on the `Categories` tab (see Figure 3 on page 21). If you want to use field codes in a standard response you must create them on the `Field Codes` tab, as described in "Field Codes" on page 44.

You can create two versions of a standard response, one in plain text format and one in HTML format (see "Filling out the HTML part tab" on page 32). When E-mail Server Java uses the standard response to create an e-mail (for example, when generating an acknowledgment), it creates a multipart e-mail that includes both plain text and HTML versions. Then the settings of the e-mail client that receives the e-mail determine which version displays.

You should be aware that e-mail clients may display multipart e-mails in varying ways. For example, if Microsoft Outlook has AutoPreview turned on, the preview may show the plain text version whereas the full display shows the HTML version. For this reason you should be careful that the plain text and HTML versions have identical content.

Note: To create and edit an HTML version of a standard response, you must have Internet Explorer 5.5 or later on the same host as Knowledge Manager.

Procedure: Creating or editing a standard response

Purpose: To create prewritten content that can be used in responding to interactions.

Prerequisites

- You must first create a category tree.
- If you want to use field codes in your standard responses, you must first create the field codes (page 48).

Start of procedure

1. On the Category Tree pane, select the category that you want to add a standard response to.
2. On the Standard Responses pane, right-click and select `New Standard Response`. The Standard Response Editor displays, as shown in Figure 8.

Note: The title of the Standard Response Editor dialog box is `New Standard Response` if you are creating a standard response and `Edit Standard Response` if you are editing an existing standard response. The contents are otherwise identical.

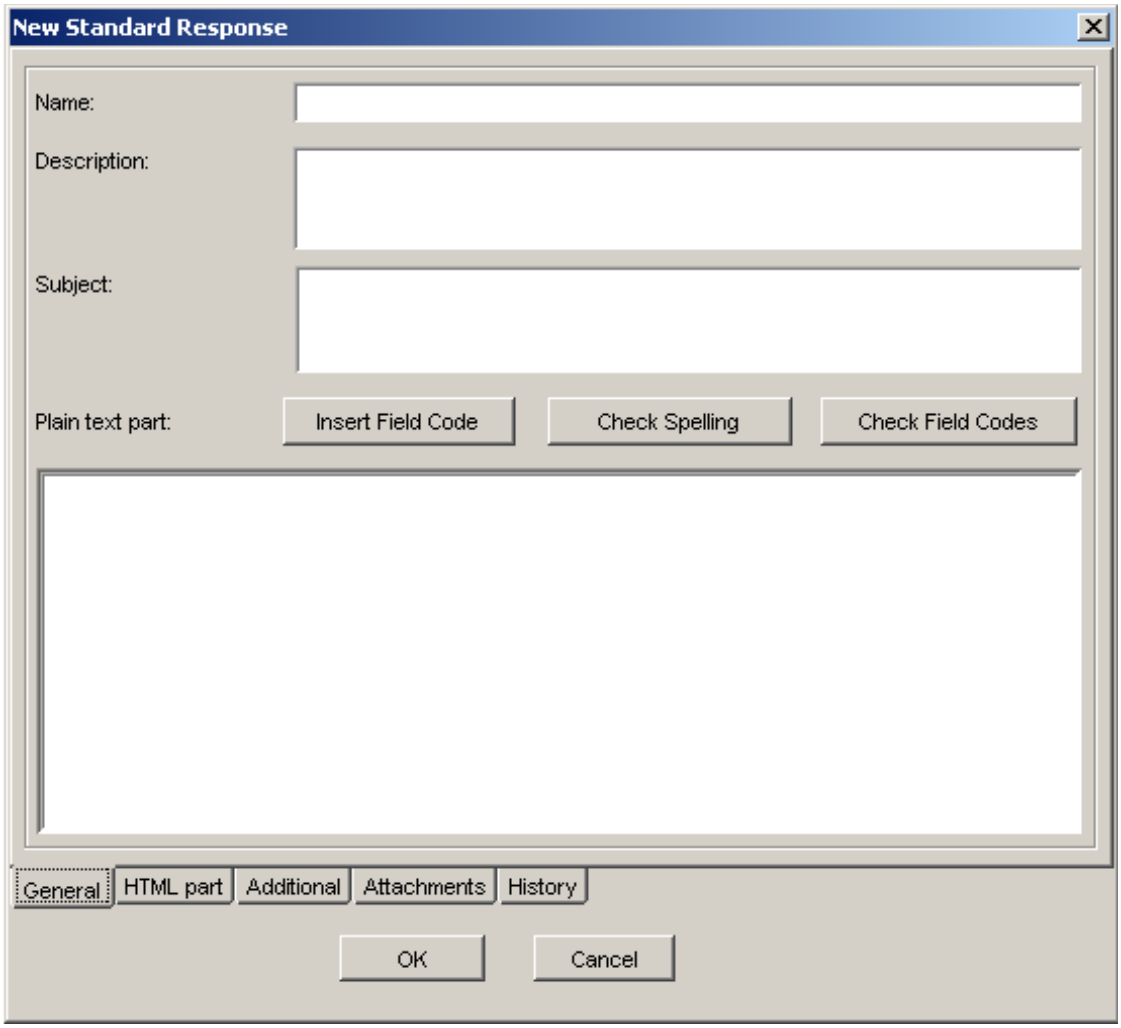


Figure 8: New Standard Response

The display opens on the `General` tab.

3. Enter a name (required) and description (optional) for the standard response. Remember that the name can use only alphanumeric characters (A–Z, a–z, 0–9), hyphen, underscore, and space.
4. Enter a subject for the standard response.

   What you enter here appears as the Subject line in an e-mail (such as acknowledgement or autoresponse) generated from this standard response.

   You can also specify a Subject line in any IRD strategy object that has a Format tab (see the *Universal Routing 7.6 Reference Manual*). If you do, this overrides any Subject line that is specified for the standard response in Knowledge Manager.
5. Enter text for the plain text version of the standard response, using the `Insert Field Code` button to insert field codes (see "Field Codes" on page 44). If you have not yet created field codes, you can continue creating the standard response, then return to it and insert field codes after you have created them.
6. Click `Check Field Codes` to show the result of applying (rendering) the field codes in this standard response. The values rendered are taken from a collection of generic properties that exists for this purpose.
7. Click `Check Spelling` to check spelling for the language selected in the adjacent drop-down list. See "Language and Dictionary Names" below for more information on this.

End of procedure

Next Steps

- Read further information about the names of languages and dictionaries (this page).
- Go on to the `HTML part` tab (page 32).

Language and Dictionary Names

For each language with spelling checker support, Knowledge Manager maintains a number of files in the `\lex` subdirectory of the Knowledge Manager installation directory. This section explains several features of the following file types:

- `<language_name>_spllchk.pro` file—Properties file, contains settings for options such as case sensitivity. Also points to the dictionary and user dictionary files.
- `userdic_<language_name>_spllchk.tx` file—User dictionary file, contains words that you add by selecting `Add` in the `Spelling` dialog box.

For a language to display in the `Check Spelling` drop-down list, it must have both of the following:

- `<language_name>_spllchk.pro` file

- A Business Attribute, of type `Language`, whose Attribute Value name is `<language_name>`, exactly matching the name of the `.pro` file.

For example, the file name `BrazilianPortuguese_spllchk.pro` shows that the name of the Business Attribute for this language must be `BrazilianPortuguese`, not `Brazilian Portuguese` or `BrazPort` or anything else.

Renaming a Language

It is not possible to rename an existing language Business Attribute. You can, however, alter the language name that displays in the `Check Spelling` drop-down.

For example, if you build a German-language category tree, you may want the name of the language to appear as the German word *Deutsch* rather than the English word *German.* So you can create a Business Attribute called `Deutsch` and select it when creating your category tree. But when it comes to checking the spelling, Genesys Knowledge Management uses the name *German,* as shown by the filename `German_spllchk.pro`. With the `.pro` file and the Business Attribute having nonmatching names, German will not be available on the drop-down list. You can rectify this situation by renaming the dictionary file:

1. Locate the file `German_spllchk.pro` and rename it `Deutsch_spllchk.pro`.
2. Refresh the view in Knowledge Manager (select `Refresh` from the `View` menu or click the refresh icon).

Saving a User Dictionary

If you have customized your spelling checking by adding items to a user dictionary, you will want to avoid overwriting your user dictionary when installing a new version of Knowledge Manager (or reinstalling the existing one).

Procedure: Saving an existing user dictionary

Start of procedure

1. Make a copy of the existing `userdic.tlx` file, located in the existing Knowledge Manager installation directory.

   Note: Prior to release 7.6.1, Knowledge Manager had spell checking for English only, so there was only one user dictionary file, named `userdic.tlx`. This section describes saving the contents of this English user dictionary.

2. Install or reinstall Knowledge Manager.

3. Locate the new `userdic_English_spllchk.tlx` (in the new Knowledge Manager installation directory) and replace its contents with the contents of `userdic.tlx`.

End of procedure

Procedure: Filling out the HTML part tab

Prerequisites

For the HTML version of a standard response, you must first create the standard response as a standalone HTML file, using whatever authoring tool or text editor you prefer.

Start of procedure

1. Start on the `HTML part` tab, as shown in Figure 9.

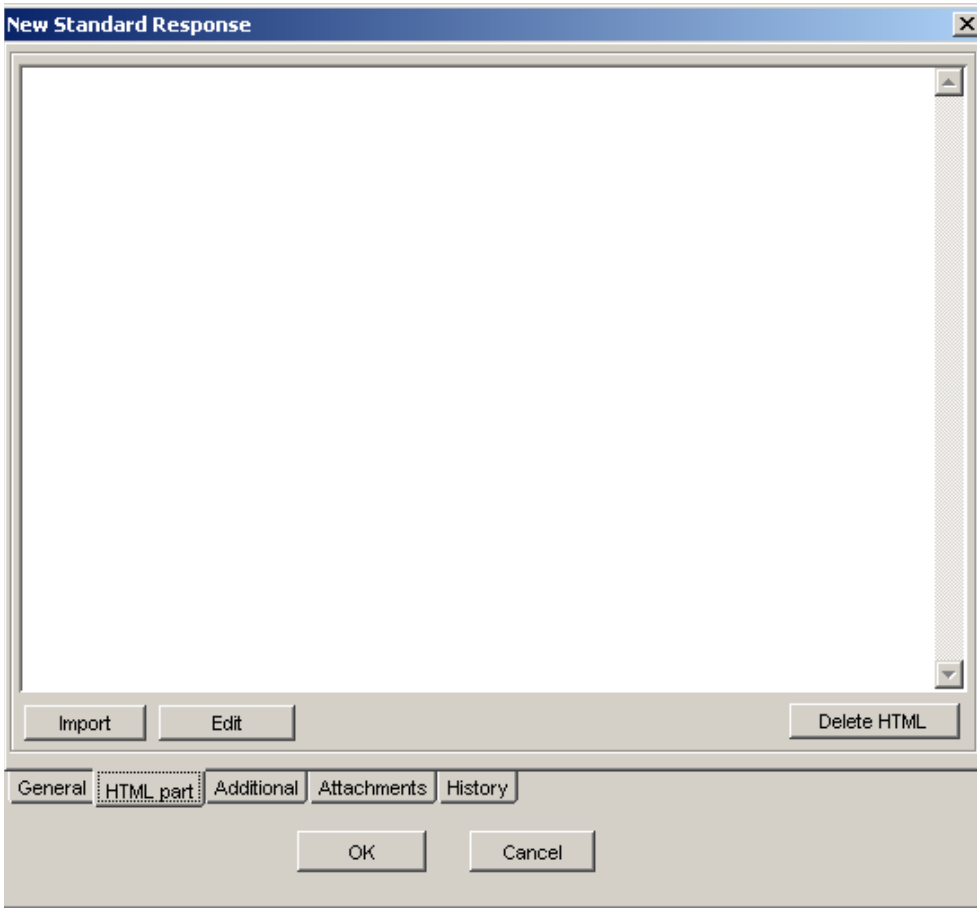


Figure 9: HTML Part Tab

2. On the `HTML part` tab, click `Import` and browse to the HTML file that you want to import, then click OK. The `HTML Standard Response` dialog box appears, as shown in Figure 10.

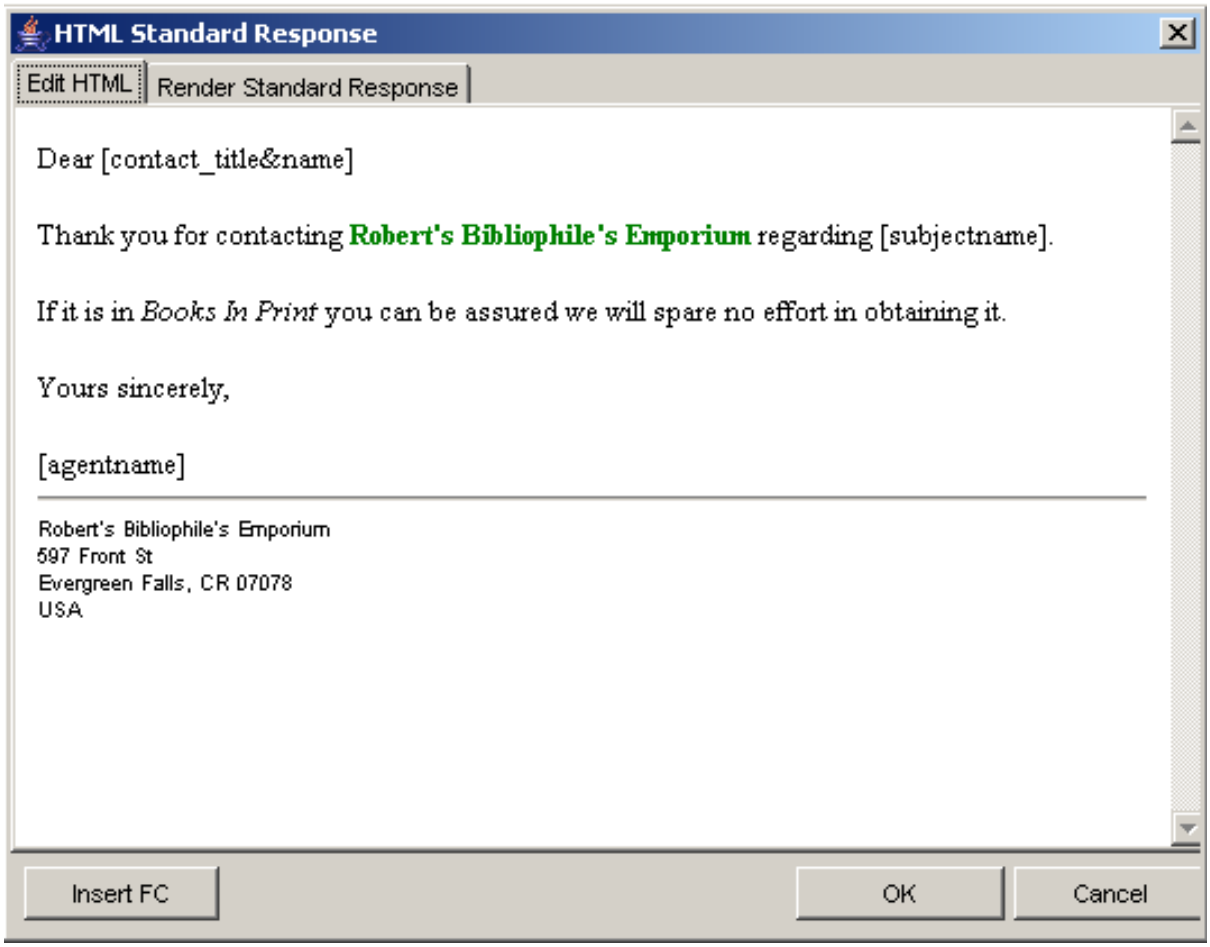


Figure 10: HTML Standard Response

Note: In the HTML version of a standard response, links to resources that are used in the content, such as image files, must use absolute URLs; embedded graphics or relative links are not supported. Also, these resources must be available on the web through an HTTP server at the time that the standard response containing this HTML is sent to the customer.

This dialog box opens with its `Edit HTML` tab. On this tab you can do the following:

- Insert, delete, copy, and paste text. Pasted text retains its attributes. Inserted text takes on the attributes of its insertion point.

- Insert field codes. Note that the imported file in Figure 10 includes bracketed expressions like [subjectname] to indicate where field codes should be inserted. Using the `Insert FC` button to insert field codes produces the result shown in Figure 11.

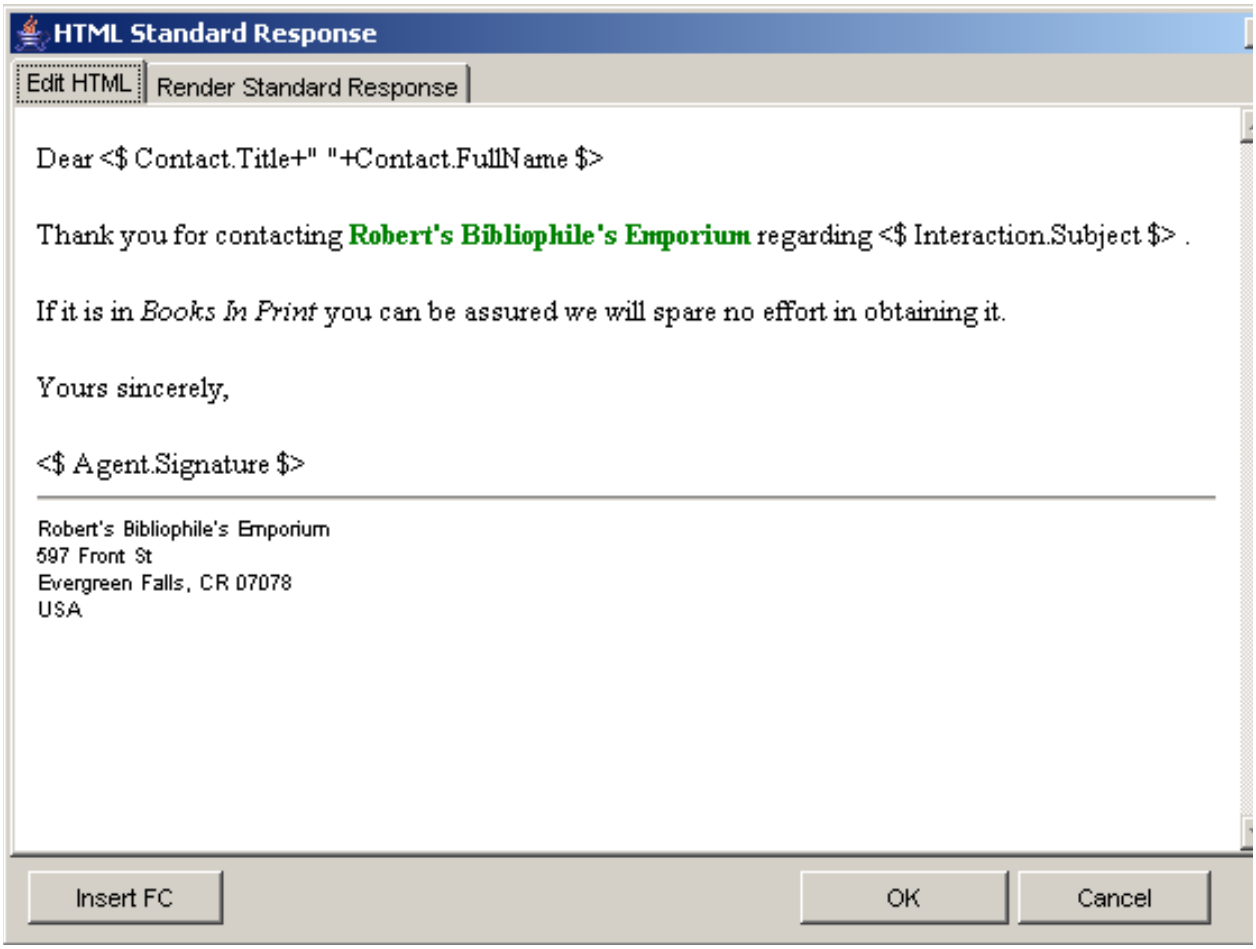


Figure 11: HTML Standard Response with Field Codes Inserted

Note: Be careful that the content of the HTML version, including field codes, matches the plain text version that you created on the `General` tab. See the explanation of the relation between plain text and HTML versions on page 28.

Go to the `Render Standard Responses` tab to see the standard response with the field codes rendered in the same way as when you click `Check Field Codes` on the `General` tab, as described in "Inserting field codes into a standard response" on page 50.

End of procedure

Next Steps

- Go on to the `Additional` tab (next section).

Procedure:
Filling out the Additional tab

Start of procedure

1. You must specify the possible uses of the standard response by using the check boxes on the `Additional` tab, shown in Figure 12 on page 35.

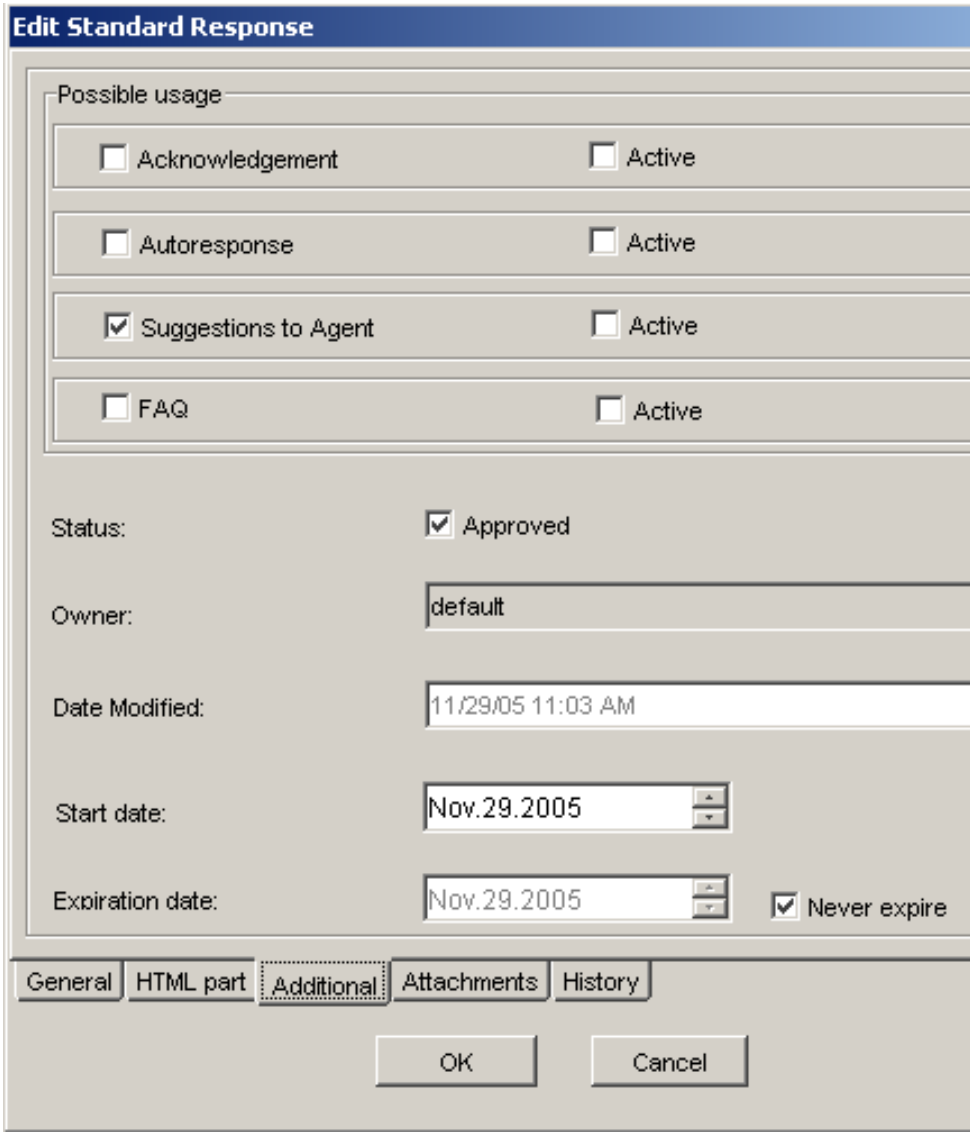


Figure 12: Standard Response, Additional Tab

The possible uses are:

   a. `Acknowledgment`—The standard response may be sent to acknowledge receipt of an incoming interaction.
   b. `Autoresponse`—The standard response may be used as an automatic response to an incoming interaction.
   c. `Suggestions to Agent`—The standard response may be offered to agents as suggested wording to use in their own replies to interactions.
   d. FAQ—The standard response may supply the answer for an item in an FAQ object. This use type applies only with Genesys Content Analyzer.

2. For each usage type, you must also specify this standard response as `Active` or not. Each category may have multiple standard responses of each usage type, but only one standard response of a given usage type can be `Active`. For more information on this point see "The Meaning of Active" below.
3. Specify the status using the `Approved` check box. Only `Approved` standard responses can appear in Routing objects.
4. Use the other check boxes and fields on this tab to specify the owner, date modified, start date, and expiration date if any.

   If a standard response's expiration date has been reached, it has the following effects:
   - The standard response is not shown in IRD, so it cannot be used in a new or modified strategy.
   - If this standard response was saved in a strategy before the expiration date was reached, E-mail Server Java does not send the standard response, but returns an error message.

End of procedure

Next Steps

- Read further information about the meaning of *Active,* on this page.
- If you want to add an attachment to the standard response, go on to the `Attachments` tab (page 37)
- If you want to maintain multiple versions of the standard response, go on the `History` tab (page 38).
- Learn how to search for standard responses (page 40).
- Create field codes to use in your standard responses, as described in "Field Codes" on page 44.

The Meaning of Active

- Purpose of `Active`—There are times when the system must select, without immediate user input, a standard response of a given usage type for a given category. If there are multiple standard responses of a single usage type for the category in question, the system selects the one that is designated as `Active`.
- Changing the `Active` standard response—If you attempt to select `Active` for a Standard Response (either a new one or an existing one), and there is already an `Active` Standard Response with that usage type for that category, Knowledge Manager offers to take the previously-`Active` Standard Response out of `Active` status, displaying the message shown in Figure 13.

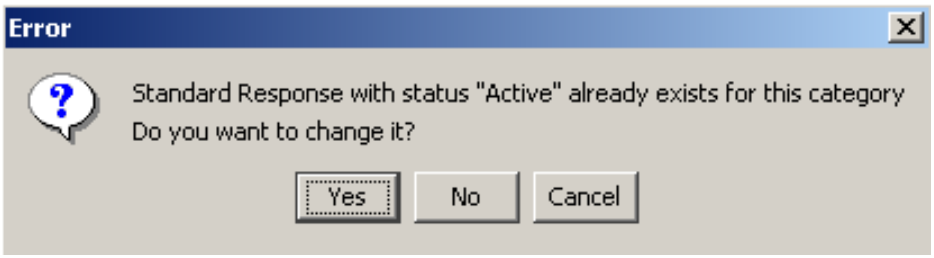


Figure 13: Changing the Active Standard Response

The Attachments Tab

The `Attachments` tab displays a list of attachments to the standard response, as shown in Figure 14.

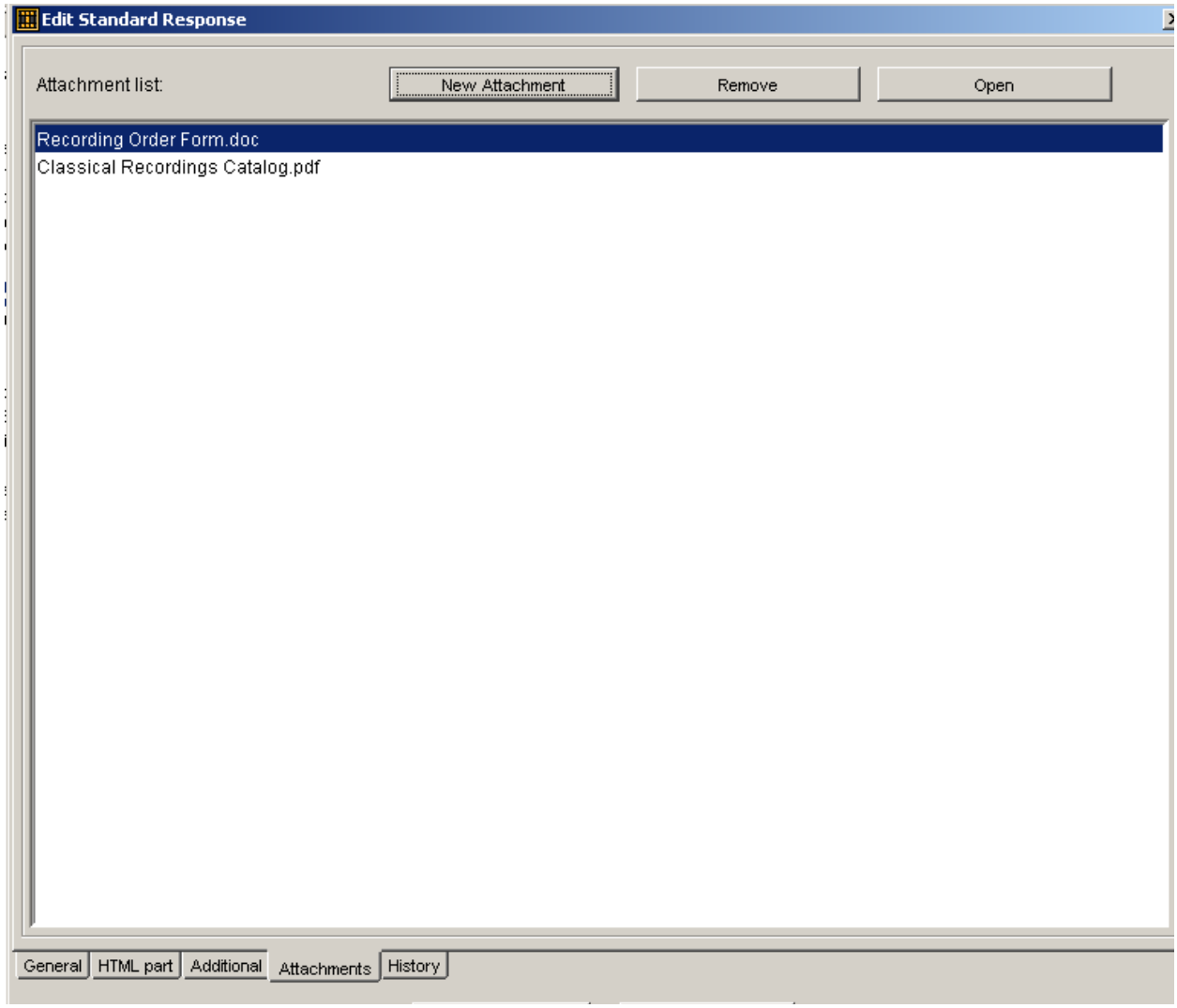


Figure 14: The Attachments Tab

On this tab you can do the following:

- To add an attachment, click `New Attachment`. Browse to the file that you want to attach, then select it.
- To remove an attachment, select it, then click `Remove`.
- To view an attachment, select it, then click `Open` (you cannot double-click on the attachment name to view it).

> **Note:** This opens the attachment for viewing only. You cannot save any changes that you make to the attachment.

The History Tab

Use the `History` tab, shown in Figure 15, for version control.

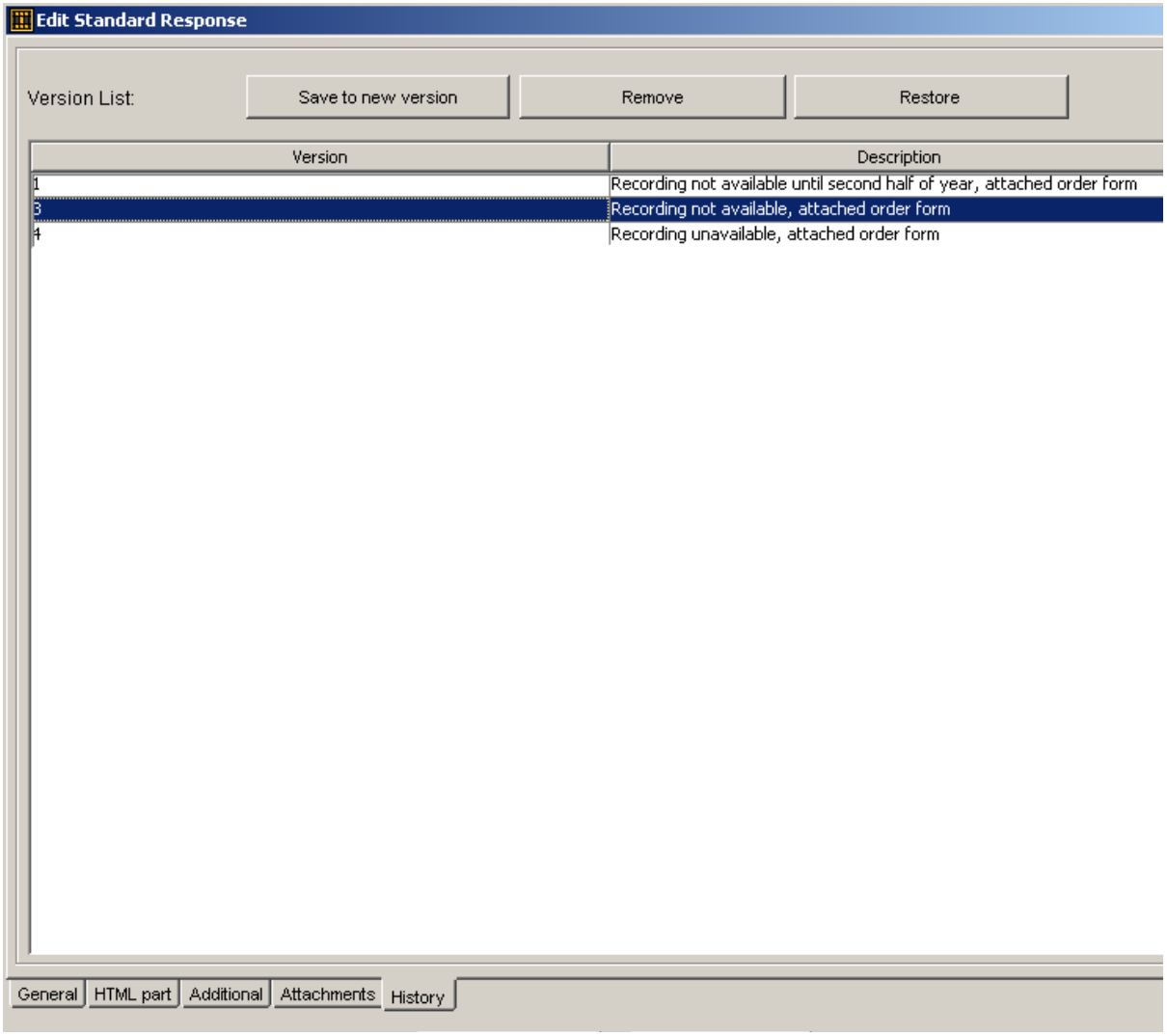


Figure 15: The History Tab

On this tab you can do the following:

- To save multiple versions of the standard response, click `Save to new version`.
- To choose the version that appears on the Standard Response pane, select the version and click `Restore`.

Restoring a version restores only the Text and Description parts of the standard response. For example:

1. Version 1 of a standard response has an expiration date of December 10, 2010.
2. Create a Version 2 with `Never expire` selected (no expiration date).
3. Restore Version 1.

4. The restored version has `Never expire` selected. The former expiration date of December 10, 2010 is not restored.

Searching for Standard Responses

You can search for the standard responses that are associated with a selected category and its subcategories.

Procedure: Searching for a standard response

Start of procedure

1. Select a category on the Category Tree pane.
2. Right-click anywhere in the Category Tree pane and select `Find Standard Responses` from the shortcut menu, as shown in Figure 16.

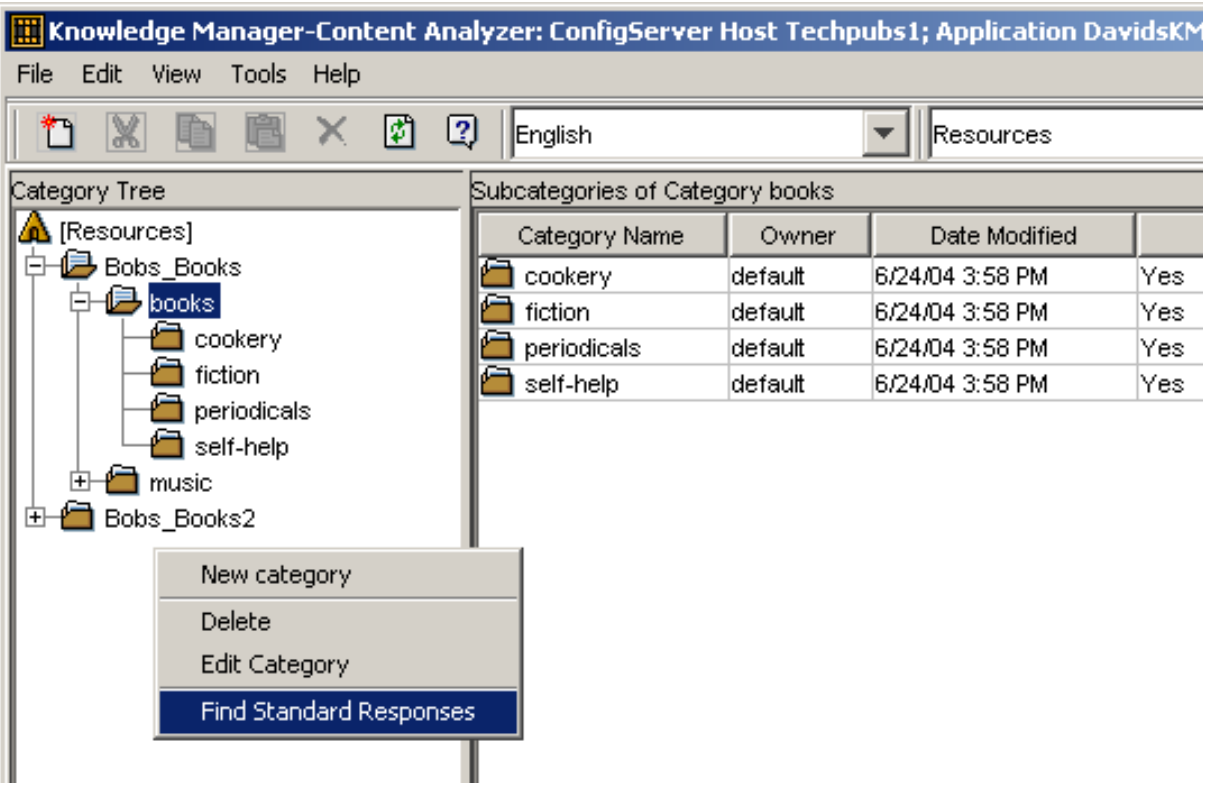


Figure 16: Find Standard Responses

The search function searches for standard responses associated with the selected category and all of its subcategories.

3. The `Find Standard Responses in Subcategories of Category: <name>` dialog box appears, as shown in Figure 17

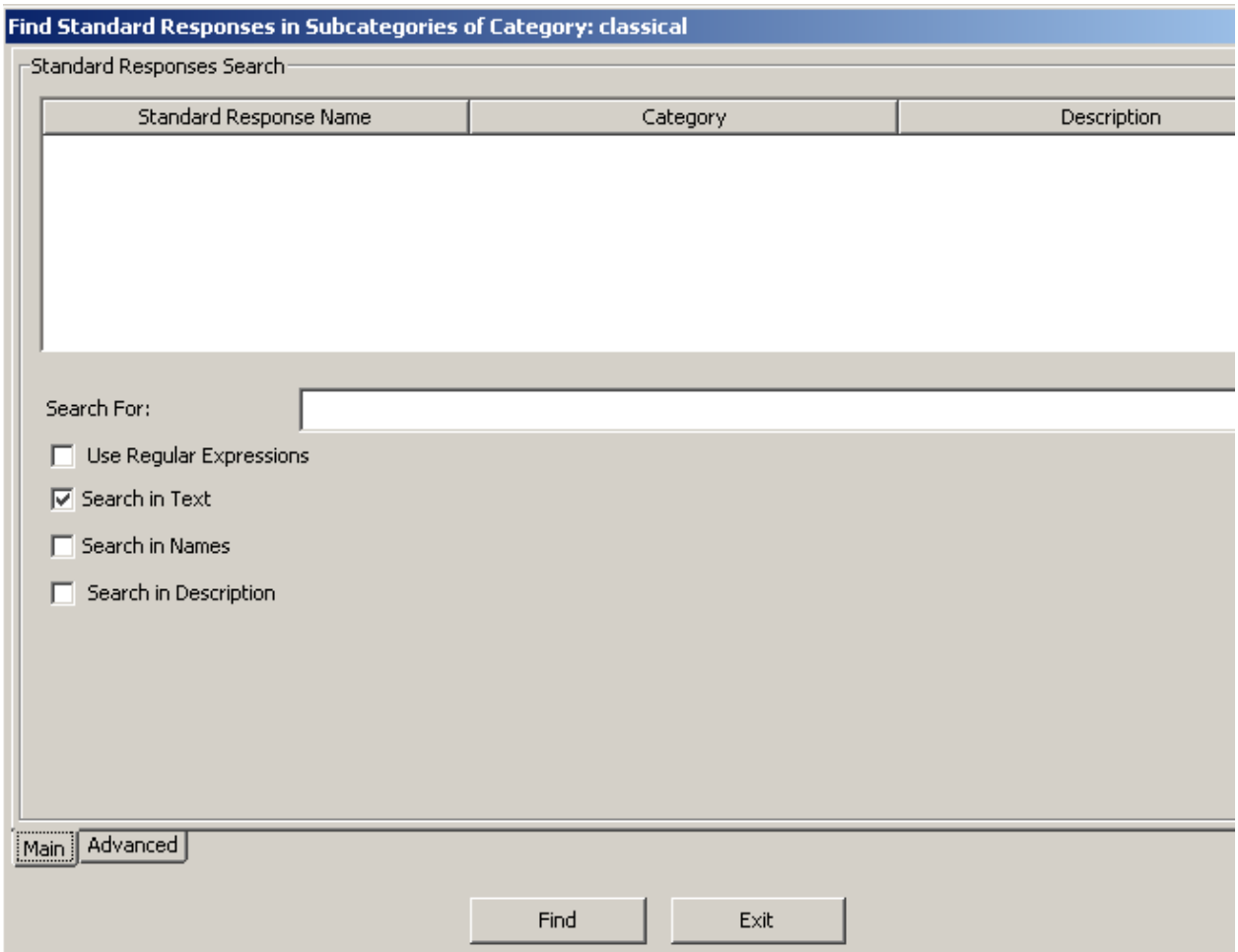


Figure 17: Find Standard Responses Dialog, Main Tab

4. Enter a string to search for in the `Search For` field (if there is nothing in this field, the system reports `Empty search pattern`). Select `Use Regular Expressions` if you want the search to treat the string as a regular expression. For more information see "Regular Expressions" on page 73.
5. Select check boxes to search in the text of the standard response, its name, its description, or any combination. The `Search in Text` box is selected by default (if none of these check boxes is selected, the system reports `No Standard Responses have been found`).

   Note: When the `Search in Text` box is selected, Knowledge Manager searches both plain text and HTML versions.

6. Go to the `Advanced` tab, shown in Figure 18, to select further attributes, in any combination.

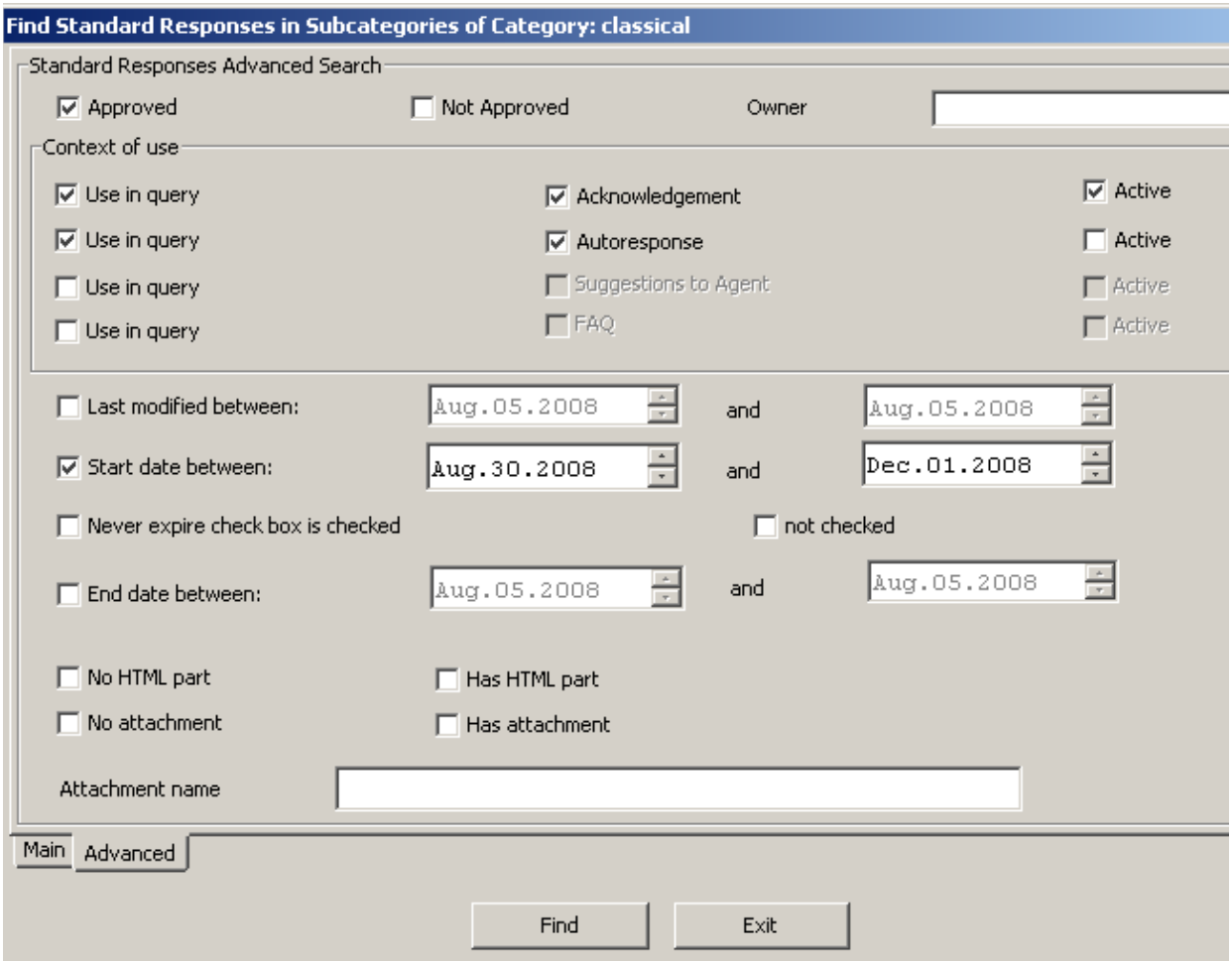


Figure 18: Find Standard Responses Dialog, Advanced Tab

7. Click `Find`. The dialog box displays the name, category, and description of all standard responses found. Click `Exit` to close the dialog box.

End of procedure

Standard Responses for SMS Gateways

Standard Responses can be used to carry the body of a Short Message Service (SMS) message that E-mail Server Java sends using an SMS gateway. To accomplish this you must use a routing strategy that includes a CreateSMS object and specifies certain attached data. For details on this strategy configuration see the "Multimedia Objects" section of the "Interaction Routing Designer Objects" chapter of the *Universal Routing 7.6 Reference Manual.*

The form of the special standard response differs according to the requirements of the gateway that you are using. This section provides examples of standard responses that can be used with three available gateways.

Clickatell

For the Clickatell gateway, create a standard response with the following as its body:

```
api_id:1234
user:Name
password:Secret
from: <$AttachedData("OrigSMSNumber")$>
to: <$AttachedData("DestSMSNumber")$>
text: <$AttachedData("SMSText")$>
```

Where:

- `1234` should be replaced with the api_id that you received upon registering for the service.
- `Name` should be replaced with the user name that you created when registering for the service.
- `Secret` should be replaced with the password that you created when registering for the service.
- The user data `"OrigSMSNumber"` contains the number of the originating SMS device.
- The user data `"DestSMSNumber"` contains the number of the recipient SMS device.
- The user data `"SMSText"` contains the text of the SMS to send (limited to 160 characters).

You can add other available parameters to the body. For information about what parameters are available for this gateway, do as follows:

1. Go to `http://www.clickatell.com/`.
2. Select `Developers`, then `SMTP`.

SMS Gateway for Mdaemon and sms2email.com

For the SMS Gateway for Mdaemon or the sms2email.com gateway, create a standard response with the following as its body:

```
<$AttachedData("SMSText")$>
```

Where the user data `"SMSText"` contains the text of the SMS to send (limited to 160 characters).

For more detailed information on Mdaemon:

1. Go to `http://www.achab.com/`.
2. Click `SMS Gateway for MDaemon`, then `Features -Outbound SMS`.

For more detailed information on sms2email.com:

1. Go to `http://www.sms2email.com/`.

2. Click `Developer Info`, then `Email to SMS Gateway`, then `how-to guide`.

Field Codes

Although field codes are used mostly in standard responses, they are the most complex and powerful aspect of standard responses, so they are described in this separate section.

Overview

The main use of field codes is to particularize standard responses, in a manner similar to the Mail Merge feature in word processors.

For example, you can use the field code `<$Contact.FirstName$>` in a response beginning `Dear <$Contact.FirstName$>`, which you send to dozens of recipients. In each message, `<$Contact.FirstName$>` is replaced by the first name of the addressee of the message (the contact) as listed in the Universal Contact Server database.

More generally, a *field code* is a formula that you insert into an e-mail that E-mail Server Java generates when triggered to do so by a routing strategy object.

The most common type of such an e-mail is standard response (triggered by an Autoresponse or Acknowledgement object), but you can also insert field codes into other types, such as chat transcripts and forwarded or redirected e-mails. In some cases the only place you can insert a field code is in the Subject line using the `Format` tab in a strategy object.

The following is a complete list of the strategy objects that can use field codes either in a standard response or in the Subject line:

- Acknowledgement
- Autoresponse
- ChatTranscript
- Create EmailOut
- Create Notification
- Create SMS
- Forward

The following is a complete list of the strategy objects that can use field codes only in the Subject line:

- Redirect
- Reply from External Resource
- Send

When an e-mail or other object containing such a formula is processed, the following happens:

1. The formula performs an operation, which produces a result.
2. The result replaces the field code in the e-mail.

This process of performing an operation and substituting its result is called *rendering*.

Note: Field codes can be used in outbound e-mails only.

Using UCS Data in Standard Responses: System Variables

In the example just given, `Contact.FirstName` retrieves a piece of data about the interaction. The ability to access interaction data is perhaps the most frequent use of field codes. Although field code formulas can be very complicated, many simply retrieve a single piece of data, such as a contact's name.

You access Universal Contact Server data using predefined variables, called *system variables*. These variables access three predefined objects. Each object has a name and a set of properties. In the example, `Contact` is an object and `FirstName` is one of its properties. The system variable `Contact.FirstName` retrieves the value of the `FirstName` property of the `Contact` object. In similar fashion, there is a system variable for each object+property pair. The objects and properties that you can use in field code formulas are described in the following sections.

Interaction

This object represents the particular interaction being worked on—that is, an inbound e-mail. These are its properties:

- `Id`
- `DateCreated`
- `Subject`
- `ToAddress`
- `FromAddress`
- `AttachedData`
- `TimeZone`

Contact

This object represents the contact associated with the interaction being worked on. These are its properties:

- `Id`
- `Title`
- `FirstName`
- `LastName`
- `FullName`

- `PrimaryPhoneNumber`
- `PrimaryEmailAddress`

Agent

This object represents the agent working on the interaction. These are its properties:

- `FirstName`
- `LastName`
- `FullName`
- `Signature`

Note: Automated responses use the default agent. Create the default agent as a Person object just like any other in Configuration Manager. Then select this Person on the `Automated Reply Agent` screen of the E-mail Server Java configuration wizard (or set this Person as the value of the `autobot-agent-login-name` option in the `E-Mail Processing` section of the E-mail Server Java application). Since this is the Person who the automated response appears to be from, you may want to name it after your company or institution.

Custom Variables

In addition to the system variables, you can use Knowledge Manager to create custom variables. Custom variables have the following properties:

- Their values are assigned by strategy objects.
- Therefore, standard responses that use field codes containing custom variables must have the usage type `Autoresponse` or `Acknowledgment`.

For an example of the use of a custom variable in a standard response, see “” on page 55. For a complete description of the Routing objects that can use custom variables, see the *Universal Routing 7.6 Reference Manual.*

Note: The names of custom variables must begin with an alphabetic character or underscore, and the remainder of the name must consist only of alphanumeric characters or underscores. This differs from the requirements for the names of other Knowledge Manager objects, which may also contain hyphen and space. For example, `5-usercode` is not an acceptable name for a custom variable, but it is acceptable as the name of a screening rule or category.

Using Your Own Data in Standard Responses

It is possible to incorporate data that you keep external to Universal Contact Server into your standard responses (including automated responses). This data

could include case numbers, account information, and so on. Remember that attached data always consists of key-value pairs.

Incorporating external data into standard responses is a two-step process:

1. Retrieve the external information and add it to the interaction as attached data. One place to do this is in a routing strategy (see *Interaction Routing Designer Help*).
2. Now that you have attached the data to the interaction, you can use the `AttachedData` property of the Interaction object to access the data and incorporate it into your standard response. The `AttachedData` property requires one argument, which is the key name. The result of the following formula is the value associated with the `OrderStatus` attached-data key:

   ```
   <$Interaction.AttachedData("OrderStatus")$>
   ```

Field Codes in Knowledge Manager

Knowledge Manager separates the task of creating field codes from the task of creating standard responses. This allows you to create complex field codes that include multiple objects, formulas, and constants (see “Complex Field Code” on page 58 for an example). You can then use these complex field codes in multiple standard responses.

Note: Standard Responses that are intended for use in FAQ objects should not contain field codes. FAQ objects contain no means of rendering field codes. See also “FAQ Objects” on page 142.

Procedure: Creating field codes

Start of procedure

1. On the `Field Codes` tab of Knowledge Manager, right-click and select `New Field Code`. The `New Field Code` dialog box (also called `Edit Field Code`) opens, as shown in Figure 19.

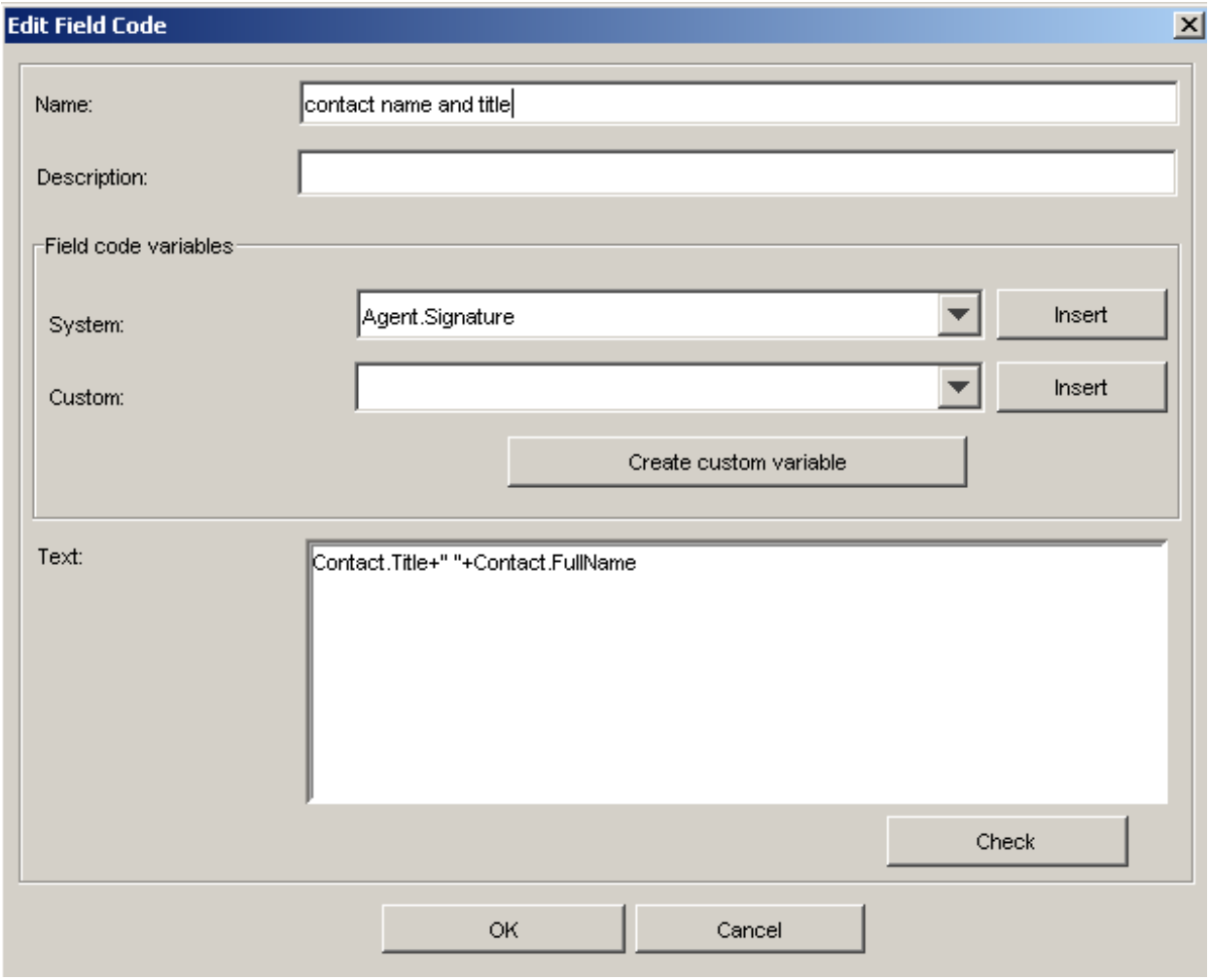


Figure 19: Edit Field Code

The `Text` field shows the field code as you create and edit it.

2. Enter a name and description for the field code. `Name` is the only required field in this dialog box.
3. The `Field code variables` section includes two fields:
   a. `System`. Click the down arrow to display a list of all system variables.

b. `Custom`. Click the down arrow to display a list of all custom variables that you have created.

To create a new custom variable, click `Create custom variable`. The `New custom variable` dialog box opens, as shown in Figure 20.

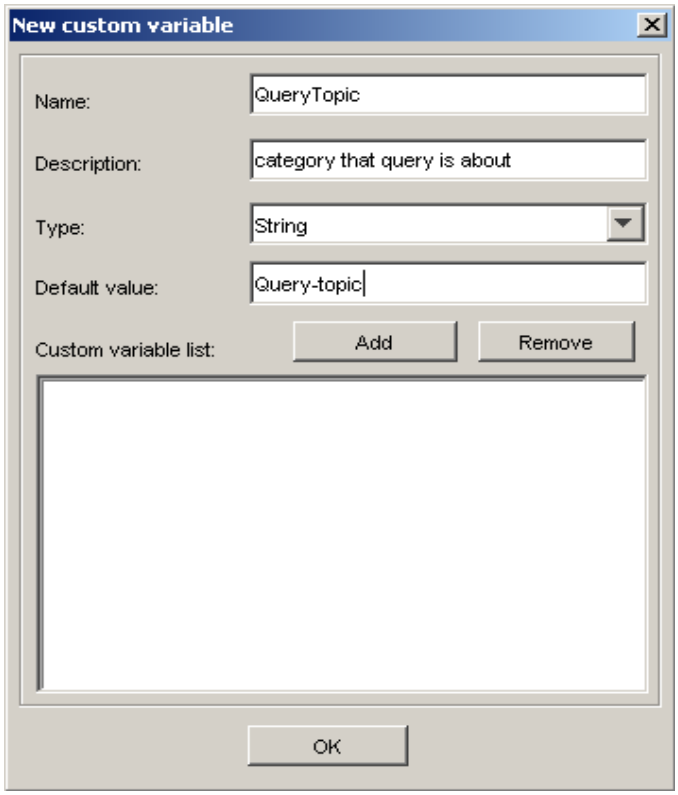


Figure 20: New Custom Variable

- Enter a name (required) and description (optional). Observe the restrictions (see page 46) on custom variable names.
- Select a type (`String` or `Integer`).
- Enter a default value (required). This value is rendered when you click `Check Field Codes` in the Standard Response Editor (see Step 4 in "Creating Standard Responses" on page 28).
- Click `Add`.

You can also use this dialog box to edit and delete existing custom variables. Click `OK` to return to the `New Field Code` dialog box.

4. Select system and custom variables from the drop-down lists, then click `Insert` to insert them into the `Text` field.

5. Enter any other desired text in the `Text` field. This text must conform to the rules described in "Field Code Syntax" on page 53.

6. Click `Check` to verify that the field code is well-formed (that is, that it has no typographical errors, missing parentheses, and so on).

End of procedure

Next Steps

- Use field codes in a standard response (next section).

Procedure: Inserting field codes into a standard response

Start of procedure

1. On the `Categories` tab, do one of the following:
   a. Double-click a standard response. The `Edit Standard Response` dialog box opens.
   b. Right-click and select `New`. The `New Standard Response` dialog box opens.

   These two dialog boxes are identical apart from their titles. Figure 21 shows `New Standard Response`.

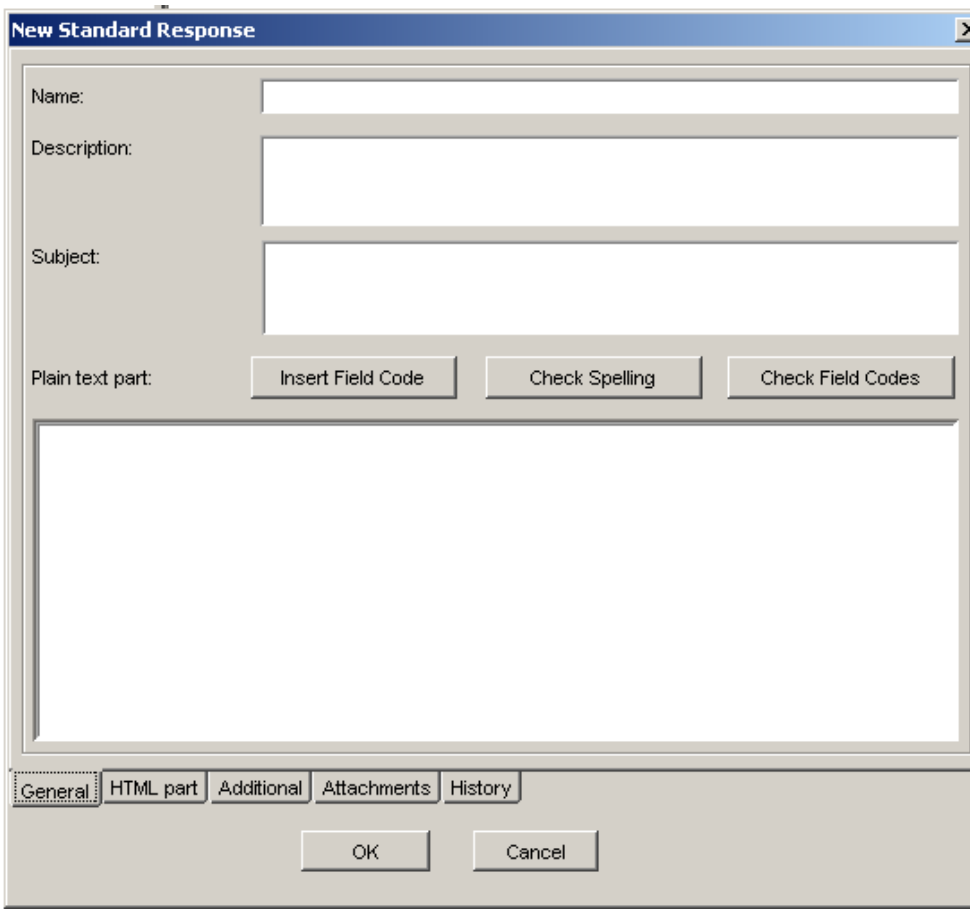


Figure 21: New Standard Response

2. Click `Insert Field Code` to display a list of all the field codes that you have created. The list appears in the dialog box shown in Figure 22 on page 52.

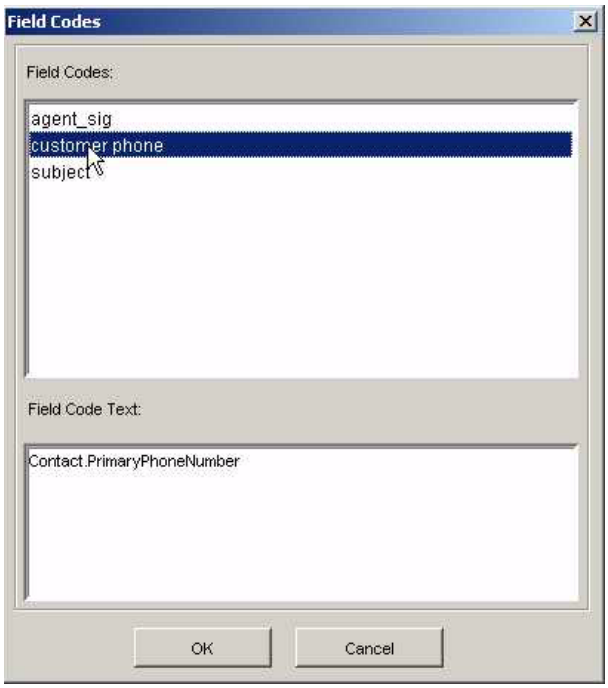


Figure 22: List of Field Codes

3. Select a field code and click `OK` to insert it, together with its required delimiters <$ $>, into a standard response. This returns you to the `Edit Standard Response` (or `New Standard Response`) dialog box.
4. Click `Check Field Codes` to see the standard response with the field codes rendered. The values rendered come from a collection of generic properties that exists for this purpose.

End of procedure

Next Steps

- Examine two examples of the use of field codes (page 55)

Field Code Anatomy

In addition to system variables such as `Contact.FirstName`, field codes may contain formulas. This section provides an outline of formula usage. For details on many of these topics, see Chapter 3 of the *Multimedia 7.6 Reference Manual*.

You must always delimit field codes by using `<$ · $>`. If you type a field code directly into the body of a standard response, then you must enter the delimiters yourself. If you select from the list of field codes in Knowledge Manager, then the delimiters are added automatically.

The text that appears inside the delimiters is a formula. Field code formulas are very similar to formulas in other applications, such as Microsoft Excel.

A *formula* is a sequence of one or more operands (such as numbers and text strings), separated by operators (such as `+` and `-`).

For example, in the following formula, `2` and `3` are operands and `+` is an operator:

```
<$2 + 3$>
```

Operands can be values that do not change (constants), or values that vary based on the context. In the previous formula, all the operands are constants, so the formula always evaluates to `5`. The next formula, on the other hand, evaluates to a different value for each agent who uses it:

```
<$Agent.Signature$>
```

Field Code Syntax

Here is a summary of field code syntax:

- As stated previously, a field code must be delimited by `<$ ... $>`.
- Alphabetic strings, whether constants in formulas or elsewhere in a field code, must be enclosed in double quotes.
- Numeric constants require no special treatment.
- You must use special characters for some purposes. For example, for your field code to render with a line break, you cannot simply type a carriage return. Instead, you must insert the expression `\n`. A list of these special characters is in Table 3 “Escape Sequences” in Chapter 3 of the *Multimedia 7.6 Reference Manual.*

Operator Precedence

If you use more than one operator in a formula, the order in which they are evaluated depends on their relative *precedence* (higher precedence operators are evaluated first). For example, multiplication (`*`) has a higher precedence than addition (`+`), so that the formula below evaluates to `14`, not `20`:

```
<$2 + 3 * 4$>
```

You can use parentheses to override the default precedence. The formula below evaluates to `20`:

```
<$(2 + 3) * 4$>
```

For a complete list of operators and their relative precedence, see “Operator Precedence” in Chapter 3 of the *Multimedia 7.6 Reference Manual.*

Data Types

Operands of several different types may appear in formulas:

- Number
- String (text)
- Date/time

- Boolean (true/false)
- Object (Contact, Interaction, and Agent)

Each data type behaves differently in formulas, and the operators have different meanings when you use them with different data types. For example, the `+` operator means "add" when used with numbers, but "concatenate" (paste together) when used with strings. This formula evaluates to *Uncle Sam Wants You*:

```
<$"Uncle Sam " + "Wants You"$>
```

In addition, some operators cannot be used with some data types at all. For example, you cannot use the multiplication (`*`) operator on two strings.

All formulas, regardless of their final data type, are converted to strings before being merged into your standard response. This conversion follows a set of default rules that depend on the data type. For example, the default rules for numbers round them off to integers. This formula causes `2` to be inserted into your standard response, even though the real result is `2.25`:

```
<$9 / 4$>
```

You can use the `Text` function (see below) or format operator (:) to override the default formatting. Either of the following formulas inserts `2.25` into your standard response:

```
<$Text(9 / 4, "#.##")$>
<$(9 / 4):"#.##"$>
```

For a detailed list of data types and how you can use them, see "Data Types" in Chapter 3 of *Multimedia 7.6 Reference Manual.*

Functions

When composing formulas, you can use many built-in functions. *Functions* are predefined formulas that perform calculations using values, called *arguments,* which you supply. To use a function, write its name, followed by an opening parenthesis, the arguments for the function separated by commas, and a closing parenthesis.

Function arguments may be of any data type, although individual functions may place restrictions on their arguments. Function arguments may be constants or formulas. The `Length` function, for example, takes a single string argument and returns its length in characters. This formula evaluates to `13`:

```
<$Length("Hello, world!")$>
```

As another example, the `Date` function takes individual date components (year, month, day, and so on), and constructs a date/time value. The formula below evaluates to `1965-11-23 09:03:10`:

```
<$Date(1965, 11, 23, 9, 3, 10)$>
```

Functions may act as arguments to other functions. The `WeekdayName` function takes a single date/time argument and returns the day of the week as a string. The formula below evaluates to `Tuesday`:

```
<$WeekdayName(Date(1965, 11, 23, 9, 3, 10))$>
```

This formula evaluates to `7`:

```
<$Length(WeekdayName(Date(1965, 11, 23, 9, 3, 10))$>)
```

For detailed descriptions of all available functions, see Chapter 3 of *Multimedia 7.6 Reference Manual.*

Using Objects

All object/property pairs are also available in the `Variables` drop-down menu in the Knowledge Manager `Field Code Editor`.

Object properties can be of any data type. `Agent.FullName`, for example, is a string, but `Interaction.DateCreated` is a date/time.

The data type of an object property can even be another object. For example, `Contact.EmailAddresses` yields another object called a `ContactEmailAddressList`. In cases such as this, you can access the properties of the resulting object by entering a period (.), followed by the property name, just as before. For example, the formula below evaluates to the number of e-mail addresses assigned to the contact:

```
<$Contact.EmailAddresses.Count$>
```

Some object properties require arguments just as functions do. For these properties, write the arguments, enclosed in parentheses after the property name, just as before. For example, the ContactEmailAddressList object has a property named `Exists`, which you can use to test whether a particular e-mail address is assigned to a contact. The data type of this property is Boolean (true/false), and it takes one argument, the e-mail address to test. For example:

```
<$Contact.EmailAddresses.Exists("samd@acme.com")$>
```

For detailed descriptions of all objects and their properties, see “Objects” in Chapter 3 of *Multimedia 7.6 Reference Manual.*

Examples

This section presents examples of the use of field codes.

Procedure: Using a custom variable

Purpose: This is a simple example of the use of a custom variable in a standard response.

Prerequisites

This example assumes a category tree that includes categories called `Cookbooks`, `Mysteries`, and `Reference`.

Start of procedure

1. In Knowledge Manager:
   a. Create a custom variable called `QueryTopic` (see "Creating field codes" on page 48).
   b. Create a field code called `Query_Topic` that consists of the variable `QueryTopic`.
   c. Create a standard response of type `Autoresponse` called `AUTO` that includes the sentence `Thank you for your inquiry about <$ Query_Topic $>`.
   d. Create the following screening rules:
      - `Cookbook: RegExFind("cook") || RegExFind("recipe") || RegExFind("food") || RegExFind("cuisine")`
      - `Mystery: RegExFind("murder") || RegExFind("crime") || RegExFind("case of the") || RegExFind("detective")`
      - `Reference: RegExFind("dictionar") || RegExFind("encyclopedia") || RegExFind("almanac")`
2. In Interaction Routing Designer, create a strategy that applies these screening rules one after the other, assigning a different value to the custom variable for each screening rule:
   a. Create a variable called `var_screen`.
   b. Create a strategy. Start the strategy with a `Screen` object. On the `General` tab of the `Screen` object, select the `Cookbook` rule.
   c. On the `Result` tab, click `Assign values of the key-value pairs`. Then under `Output values` select `var_screen` for `Variable` and enter `ScreenRuleMatch` for `Key from output`.
   d. Connect the `Screen` object to a `Generic Segmentation` object. Create two segments: `var_screen = true` and `var_screen = false`.
   e. Connect an `Autoresponse` object to the top green port (the one corresponding to `true`) of the Segmentation object. In the Autoresponse, select the `Select standard response` radio button and select `AUTO` in the associated drop-down list.
   f. Still in this `Autoresponse`, go to the `General` tab and in the `Field Codes` area (bottom of the tab) click the `New` icon, enter `QueryTopic` under `Key`, and enter `cookbooks` under `Value`.

      This will generate an e-mail that includes the sentence *Thank you for your inquiry about cookbooks.*
   g. Return to the `Generic Segmentation` and connect a new `Screen` object to its second green port (the one corresponding to `false`).

h. On the `General` tab of the new `Screen` object, select the `Mystery` rule. On the `Result` tab, click `Assign values of the key-value pairs`. Then under `Output values` select `var_screen` for `Variable` and enter `ScreenRuleMatch` for `Key from output`.

i. Proceed as in Steps d–f: Connect this `Screen` object to a new `Generic Segmentation`, again with segments for `var_screen = true` and `var_screen = false`.

j. As in Step e, connect a new `Autoresponse` object to the green port for `true`, select the `AUTO` standard response, and enter `QueryTopic` under `Key`. This time enter `mysteries` under `Value`.

k. Return to the second `Generic Segmentation's` green port for `false` and repeat Steps g–j, creating a third `Screen` object and `Generic Segmentation`. In the `Screen` object, select the `Reference` rule; in the `Segmentation` object, set the custom variable to `reference`.

End of procedure

Figure 23 shows the strategy as described. The single standard response `AUTO` generates three e-mails, each with a different word filling the blank in *Thank you for your inquiry about ___*.

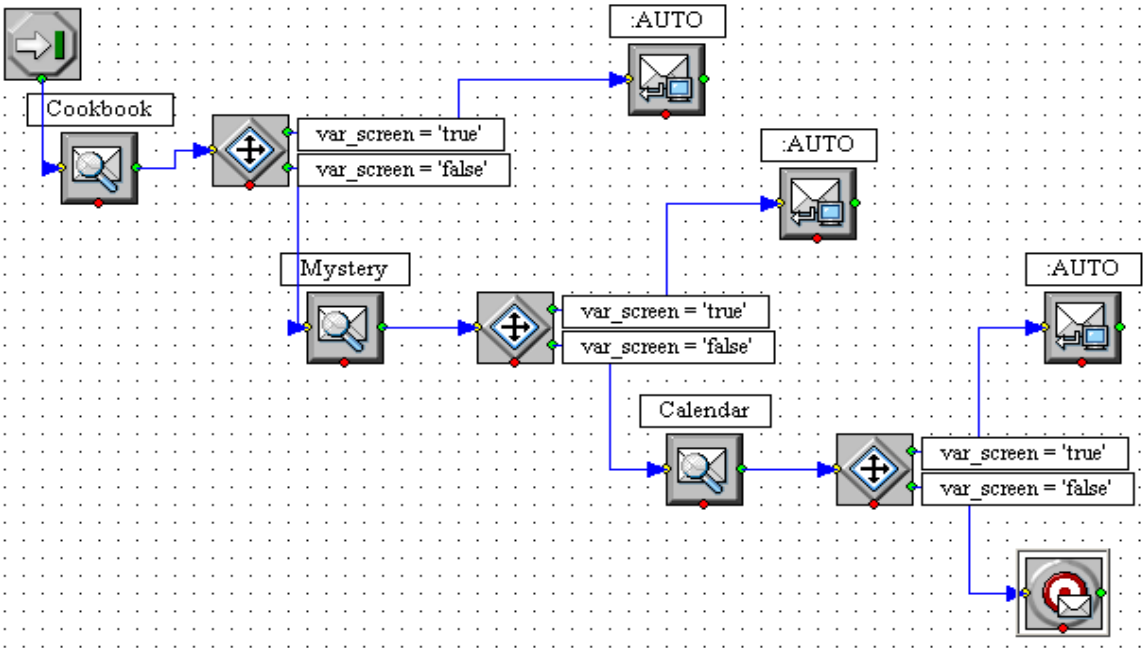


Figure 23: Strategy Using Custom Variable

Next Steps

- Examine an example of a complex field code (page 58).
- Go on to create and manage screening rules (page 58).

Complex Field Code

```
<$ If (Time() - Interaction.DateCreated > 14, "Please accept our
apologies for not having replied sooner. ", "") $>
```

This field code inserts a tardiness apology if more than 14 days have elapsed since the interaction first entered the system. It uses the function `If`, which has these properties:

- Its syntax is `If (Boolean, TrueResult, FalseResult)`
- If `Boolean` evaluates to `True`, it returns the second argument.
- If `Boolean` evaluates to `False`, it returns the third argument.

In this example the three arguments of `If` are as follows:

1. `Time() - Interaction.DateCreated > 14` A formula that returns `True` if the difference between the date created and the current system time is more than 14 days. (The result of a mathematical operation on dates is given in days.)
2. `"Please accept our apologies for not having replied sooner. "` A text string apologizing for tardiness, inserted if the formula evaluates to `True`.
3. The null string: if the reply is not late (the formula evaluates to `False`), nothing is inserted in it.

Screening Rules

Screening rules scan an interaction and try to match either a destination address, a regular expression, or both. Screening is performed by Classification Server when it is triggered by a `Screen` object in a routing strategy. A screening rule can optionally be associated with a category.

Note: Screening, like classification (Chapter 2, "Genesys Knowledge Management: Content Analyzer," on page 83) can operate on any interaction that has text somehow associated with it, whether as the body of the interaction (e-mail, chat), or otherwise (as user data, for example). In practice, it is expected that most interactions that are screened or classified will be e-mail messages; therefore this Guide uses the terms *e-mail* and *message* interchangeably to refer to these interactions. In fact whatever is said here about e-mail applies to any interaction that has associated text.

Screening Rule Editor

The `Screening Rule Editor` enables you to compose and test screening rules.

Procedure: Creating a rule

Purpose: To create or edit a screening rule.

Start of procedure

1. Go to the `Screening` tab.
2. Do one of the following:
   - Select `New` from the `File` menu.
   - Right-click anywhere on the `Screening` tab and select `New`.
   - Select an existing screening rule and double-click or select `Edit` from the `File` menu.

The `Screening Rule Editor` opens, as shown in Figure 24 on page 60. If you are creating a new screening rule, the title of the dialog box is `New Screening Rule`; if you are editing an existing rule, its title is `Edit Screening Rule`.

Edit Screening Rule

Name: Wrong transaction amount

☑ Enabled Order: 10

Use these addresses

customer_support
tech_support
warranty_support

Add

☐ Screen mailbox ☐ Exact address match

◉ AND ○ OR

Use pattern

Choose and add regular expressions and operators between them:

Find("") Add && Add

Find("transaction",true) && Find("amount",true) && (Find("wrong",true) || Find("different",true) || Find("doesn't match",true))

Search for pattern in message's ☑ Subject ☑ Body ☑ Header

☐ Merge sources checked above

Categories

Add
Delete

Root category	Category	Relevancy
Financial service	Errors in transactions	75

Test against whole database

Test messages

Add new
Delete
Test rule

Subject

Figure 24: Screening Rule Editor

3. Create or edit a name for the rule, observing the limitations on Knowledge Manager names described in the Note on page 26.
4. Use the `Enabled` check box to enable and disable the rule. The rule must be enabled in order to be available when you add a `Screen` object to a routing strategy. However, once a strategy includes a `Screen` object that uses a particular screening rule, disabling the rule does not prevent the strategy from using the rule.
5. Use the `Order` box to specify the order in which you want this rule to apply with respect to other screening rules. This ordering applies only in a Multiscreen routing object in which `All rules` is selected.
6. In the `Use these addresses` area, select an address from the left-hand window, then click `Add` to copy it to the right-hand window.
   - With the `Exact address match` box cleared, the rule looks for messages having that address as a substring of their destination address. For example, `xyz@domainname.com` matches `abc.xyz@domainname.com` and `xyz@domainname.com`.
   - With the `Exact address match` box selected, the screening rule looks for messages having that exact address as a destination. For example, `xyz@domainname.com` matches `xyz@domainname.com` but not `abc.xyz@domainname.com`. This match is not case sensitive.

Notes: The e-mail addresses listed in the left-hand window of `Use these addresses` are defined in this configuration database object: `Business Attributes > EmailAccounts > Attribute Values > Annex > general > address`.

Instead of moving addresses from the left-hand window to the right-hand, you can directly type an address in the right-hand window.

You may leave the right-hand window empty, in which case the rule ignores the address in matching.

7. Still in the `Use these addresses` area, select `Screen mailbox` to make the rule match the POP box from which the e-mail entered the Multimedia system, rather than the `To` field of the e-mail itself. The difference is that each e-mail enters the system from exactly one mailbox, while the `To` field can contain multiple addresses.

Note: For this feature to work as expected, the E-mail Server Java `enable-same-mail-from-mailboxes` option must be set to `true`. With this setting, E-mail Server Java creates a separate interaction for each address in the `To` field (that is, for each mailbox that it pulls the e-mail from when it creates the interaction).

8. Select `AND` or `OR` radio buttons:

- `AND`—Match the addresses selected in `Use these addresses` *and* match the pattern defined in `Use pattern` (see Step 9).
- `OR`—Match the addresses selected in `Use these addresses` *or* match the pattern defined in `Use pattern` (see Step 9).

9. In the `Use pattern` area, compose the rule, using the drop-down lists for functions and operators. See the next section "Syntax and Semantics" for an explanation of how to construct rules.

10. Use the check boxes to have the screening rule apply to the message body, subject, header, or any combination. You must select at least one.

With multiple check boxes selected, there are two ways that a screening rule can behave. Use the `Merge sources checked above` checkbox as one way to control this; see "Subject, Body, and Header" on page 70 for an explanation.

11. In the `Categories` area, associate the screening rule with a category:

a. Click `Add`. The `Choose category` dialog box appears, as shown in Figure 25.

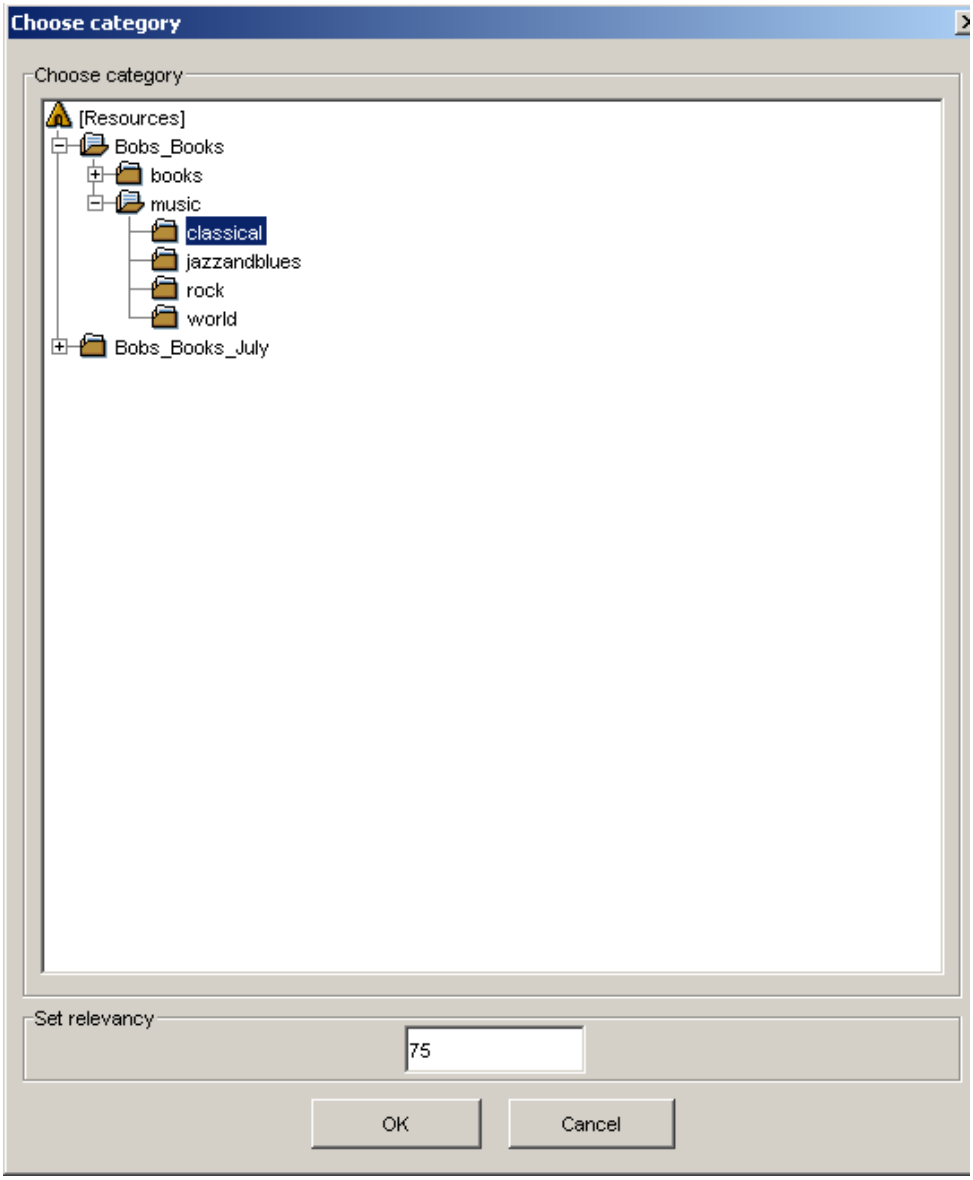


Figure 25: Choose Category

When an interaction matches this screening rule, the results are similar to classifying it using Genesys Content Analyzer: it receives a category name and a confidence level. The confidence level indicates that the system is x percent confident that this interaction belongs in this category. With Content Analyzer, the system assigns the category and confidence level. But with a screening rule, it is up to you, the user, to decide what category the interaction belongs to and how confident you are of that categorization.

Do so as follows:

b. Select the category you want to associate the screening rule with.

c. Set the relevancy in the `Set relevancy` box.

End of procedure

Next Steps

- Test the rules, as described in the next section.

Testing a Rule

There are two ways to test a screening rule: on the UCS database, and on specially-created text.

Procedure: Testing screening rules on the UCS database

Purpose: To test a screening rule on all interactions in the UCS database that relate to the current tenant.

Start of procedure

1. Click `Test against whole database` near the bottom of the `Screening Rule Editor`. The `Test screening rule against message database` dialog box appears, as shown in Figure 26.
2. Adjust the contents of the `Start date` and `End date` boxes to cover the desired timespan, then click `Find`.

   If there are a great many interactions and the testing process is taking too long, you can click `Stop`.

The results of the test display as follows:

- The `Matched messages` area lists all interactions that match the rule.
- The `Subject`, `Message`, and `Header` boxes show the text of the subject, body, and header of the interaction that is selected in the `Matched messages` area.
- The `Matched Key-Values` area shows all keys in the interaction's user data whose values match the rule. In the example in Figure 26, the screening rule created a key-value pair to hold the matched string, in this case a series of sixteen digits that could be a credit card number.

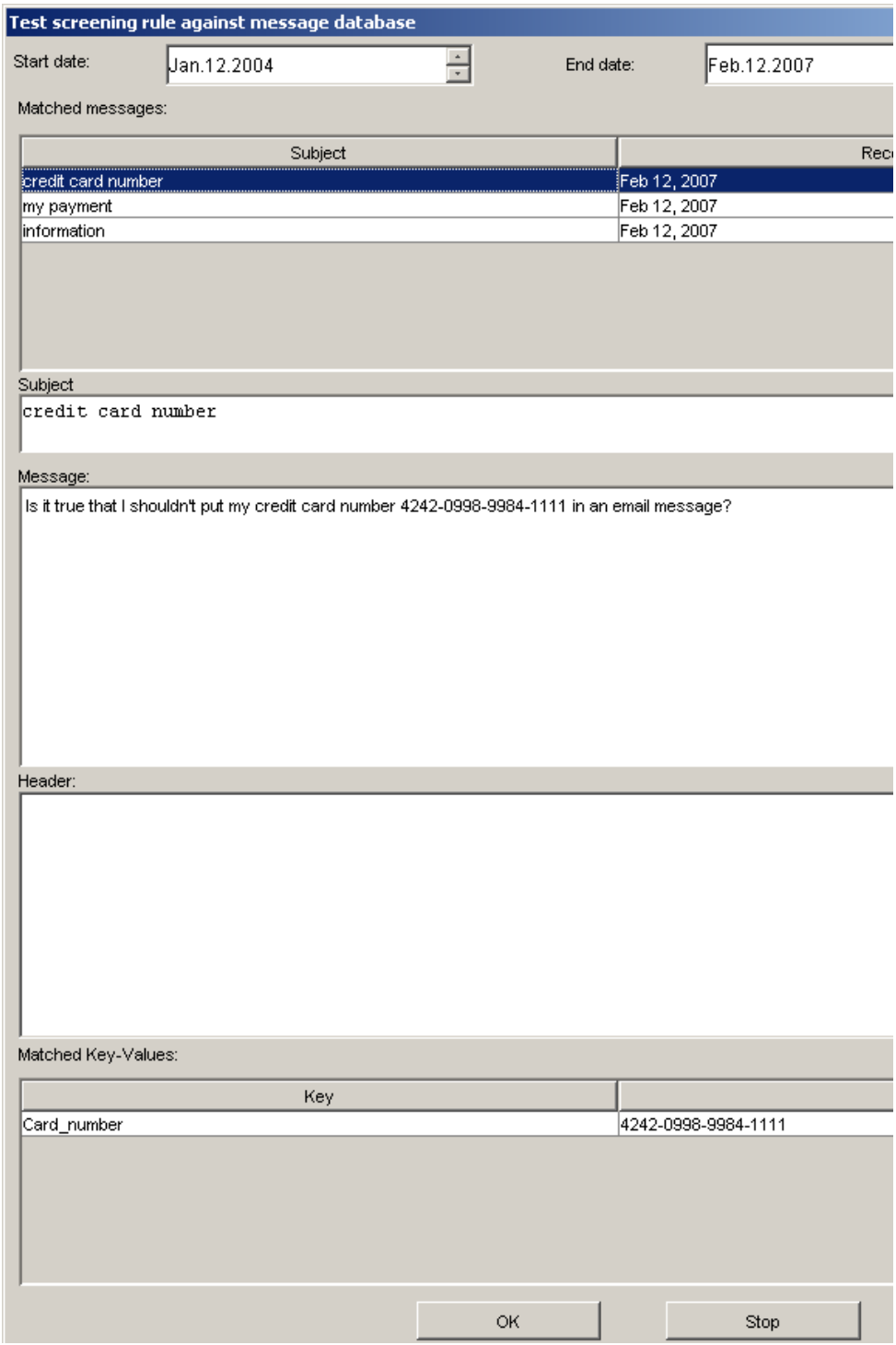


Figure 26: Test Screening Rule Against Message Database Dialog Box

End of procedure

Next Steps

- Test a rule on text that you create.

Procedure:
Testing screening rules on specially-created text

Purpose: To test a screening rule on text that you create for that specific purpose.

Summary

At the bottom of the `Screening Rule Editor` is the `Test messages` area, shown in Figure 27. Use this area to create messages that you can test screening rules on.

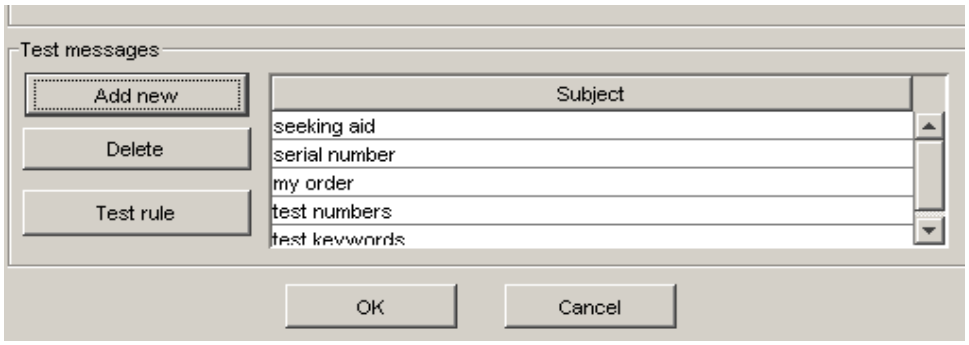


Figure 27: Test Messages Area

Start of procedure

1. Click `Add new` to open the `Test message` dialog box, in which you can enter the address, subject, and body of a test message. See Figure 28.

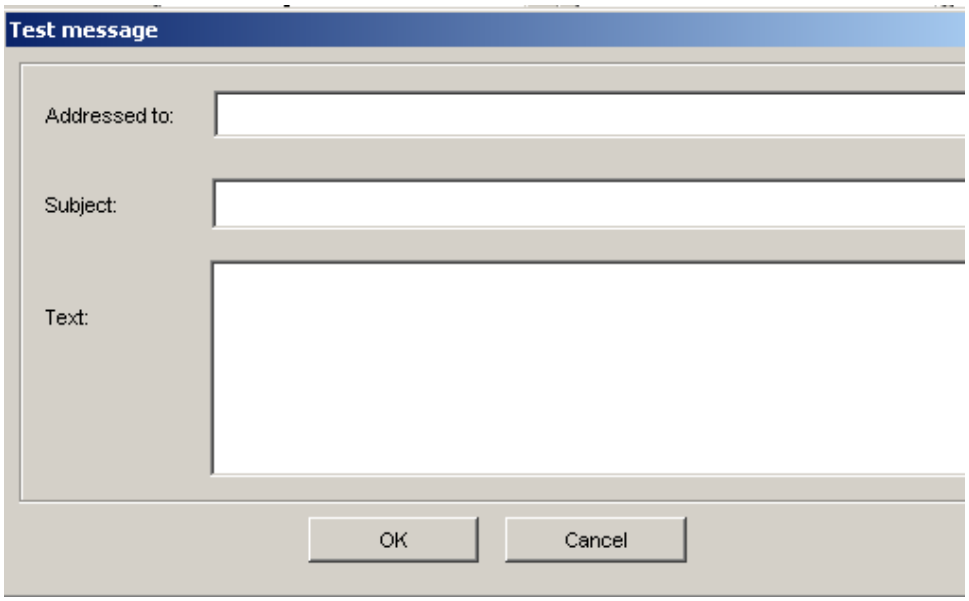


Figure 28: Test Message Dialog Box

   Click `OK` to return to the `Screening Rule Editor`.

2. Manage existing test messages, as follows:
   - Double-click an existing test message to edit it.
   - Select a test message and click `Delete` to delete it.
3. To test a rule:
   a. Select the rule on the `Screening` tab, then open the `Screening Rule Editor` as described in Step 2 on page 59.
   b. Select a test message from the list in the `Test messages` area.
   c. Click `Test rule` to apply the rule to the selected message.
   d. A message window appears, reporting whether the selected message matches the rule.

End of procedure

Next Steps

- Learn how to search for screening rules (next section).

Procedure: Finding a rule

Start of procedure

1. Right-click anywhere on the `Screening` tab, then select `Find Screening Rules`. A dialog box with this name displays, as shown in Figure 29.

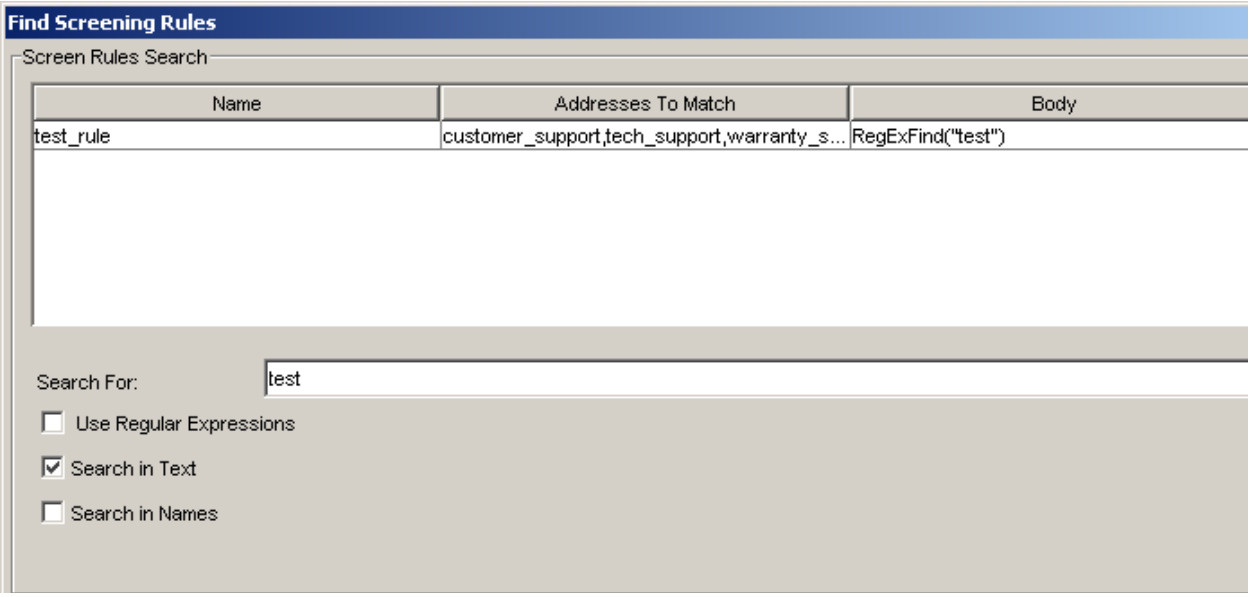


Figure 29: Find Screening Rule

2. Enter a string to search for in the `Search For` field.
3. Select `Use Regular Expressions` if you want the search to treat the string as a regular expression.
4. Select check boxes to search in the screening rule's text, name, or both.
5. Click `Find`. The dialog box displays the name, addresses to match, and body of all screening rules found.

End of procedure

Next Steps

- Read the rest of this “Screening Rules” section to learn more about how screening rules work.
- Go on to learn how to import and export Knowledge Manager objects (page 78).

What Screening Rules Check

Screening rules check the following parts of an interaction, depending on what you select in the Screening Rule Editor, and on the settings in the IRD screening objects:

- The subject, if you select that check box.
- The body, if you select that check box.
- The header, if you select that check box.

  See also "Subject, Body, and Header" on page 70 on how screening rules behave if two or more of the preceding are selected.
- The destination address, if you have put anything in the right-hand box in the `Use these addresses` area of the `Edit Screening Rule` dialog box.
- The value of any key in the user data, if both of the following are true:
  - In the Multiscreen or Classify strategy object, you select a key in the `User data key if specified` drop-down list under `Get screened data from.`
  - In the Screening Rule Editor, you select the `Body` check box in the `Use pattern` area of (see Step 9 on page 62).

  User data is first associated with the interaction by the media server when it creates that interaction. As an example, E-mail Server Java associates the following user data with the interaction:
  - FirstName (from Contact information)
  - LastName (from Contact information)
  - Mailbox (value of the `address` option in the `pop-client` section of the E-mail Server Java Application object)
  - To (MIME header field)
  - Subject (truncated to 512 characters)
  - FromAddress (personal part of From header field)
  - FromPersonal (e-mail address part of From header field)
  - All Header fields (except Received, Return-Path, X-MIMETrack, Subject, Sender, From, To, Cc, Bcc) prefixed by `Header_`
  - All parent attached data (originally created by E-mail Server Java) which can be inherited; that is, all parent attached data:
    - not starting with `Header_`
    - not starting with _ (underscore)
    - not equal to `GEM_Failure`
    - not equal to `GEM_FailureMsg`
    - not equal to `GEM_FailureArgsl`

  User data may then be added or modified by a routing strategy.

Subject, Body, and Header

If you select more than one of the Subject, Body, and Header areas, a screening rule can behave in the following two ways:

The default behavior is for the rule to apply to each area in turn; for example, with `Subject` and `Body` selected, the rule applies first to the Subject, then to the Body.

The alternative behavior is for all selected items to first be concatenated so that the rule applies to all at once. There are two ways to achieve this alternative behavior.

- To enforce it for all screening rules, set the `subject-body-header` option for both Knowledge Manager and Classification Server to `true`.
- To enforce it for a particular screening rule:
  a. Leave `subject-body-header` set to `false`.
  b. Open the rule in the Screening Rule Editor.
  c. Select the `Merge sources checked above` check box.

Note: Setting `subject-body-header` to `true` overrides any selection of the `Merge sources checked above` check box for a particular rule.

Syntax and Semantics

This section describes the structure and meaning of screening rules.

Functions and Arguments

Functions

Screening rules can use three basic functions:

- `Find("<text>")`, where `<text>` is a text string. It returns the result `true` if the interaction contains the exact string between quotes, ignoring case.
- `RegExFind("<regular expression>")`, where `<regular expression>` is a regular expression (see "Regular Expressions" on page 73). It returns the result `true` if the interaction contains any string that matches the regular expression between quotes.
- `RegExMatch("<regExp>")`, where `<regular expression>` is a regular expression. It returns the result `true` only if the entire content of the interaction matches the regular expression between quotes.

Note: `RegExFind` and `RegExMatch` are the same except that `RegExFind` looks for a match anywhere in the body of the interaction, whereas `RegExMatch` demands that the entire body of the interaction match the regular expression.

Arguments

All functions have one required argument, which must appear between double quotation marks, as represented above (`<text>` or `<regular expression>`). This required argument can be followed by one or two optional arguments, depending on the function. The full form of each function, including all arguments, is as follows:

```
Find("<text>", <IgnoreCase>)
RegExFind("<regular expression>",<"key">,<IgnoreCase>)
RegExMatch("<regular expression>",<IgnoreCase)
```

IgnoreCase The IgnoreCase argument must be a Boolean value (*true* or *false*). All three functions ignore case in searches unless you include the IgnoreCase argument with a value of `false`.

For example:

- `Find("pacific")` finds *Pacific* and *pacific*.
- `Find("Pacific",false)` finds *Pacific* but not *pacific*.

You can also substitute `true` for `false`—for example, `Find("Pacific",true)`—which means that case is ignored. So `Find("Pacific",true)` is the same as `Find("Pacific")`.

Key The `key` argument must be a string. If this argument is present, the system creates a key-value pair with the following characteristics:

- The key name is the string specified by the `key` argument, prefixed by `ScrKey_`.
- The value is the material that the screening rule matches.

The system then adds this key-value pair to the interaction's attached data.

For example, `RegExFind("[A-Z]\d\d\d","ID_code",false)`:

1. Finds strings consisting of a capital letter followed by three digits (see "Regular Expressions" on page 73).
2. Attaches to the interaction a key-value pair called `ScrKey_ID_code` whose value is A123, X005, M999, or whatever the function found in this interaction to match the regular expression.

Operators

Operators are of two types:

- Binary operators join two functions.
- Unary operators operate on a single function.

`&&` is the binary operator *and*. For example,

```
Find("interest rate") && Find("APR",false)
```

matches a message only if it includes both *interest rate* and *APR*.

|| is the binary operator *or*. For example,

```
Find("station wagon") || Find("convertible")
```

matches any message that includes either *station wagon* or *convertible* (or *Station Wagon* or *station Wagon* or *Convertible*).

`!` is the unary operator *not.* For example,

```
!Find("windows")
```

matches any message that does not include the word *windows.* You can combine `!` with a binary operator. For example,

```
Find("bird") && !Find("goose")
```

matches any message that includes *bird* but does not include *goose.*

Operator Precedence

`p && q || r` is parsed as `(p && q) || r`. For example, consider:

```
Find("debt") && Find("income") || Find("profit")
```

To paraphrase, this screening rule is basically "find X or find Y," where X is *debt* and *income,* and Y is *profit.* It matches both *debt exceeds income* and *profits are fantastic*. You can modify the default precedence by the explicit use of parentheses; for example:

```
Find("debt") && (Find("income") || Find("profit"))
```

This screening rule is basically "find X and find Y," where X is *debt* and Y is either *income* or *profit.* It matches both *debt exceeds income* and *debts impact profit.*

Pattern Builder

Knowledge Manager includes a pattern builder that offers the choice of each function type in all possible forms, with and without optional arguments, for a total of eight, as shown in Figure 30.

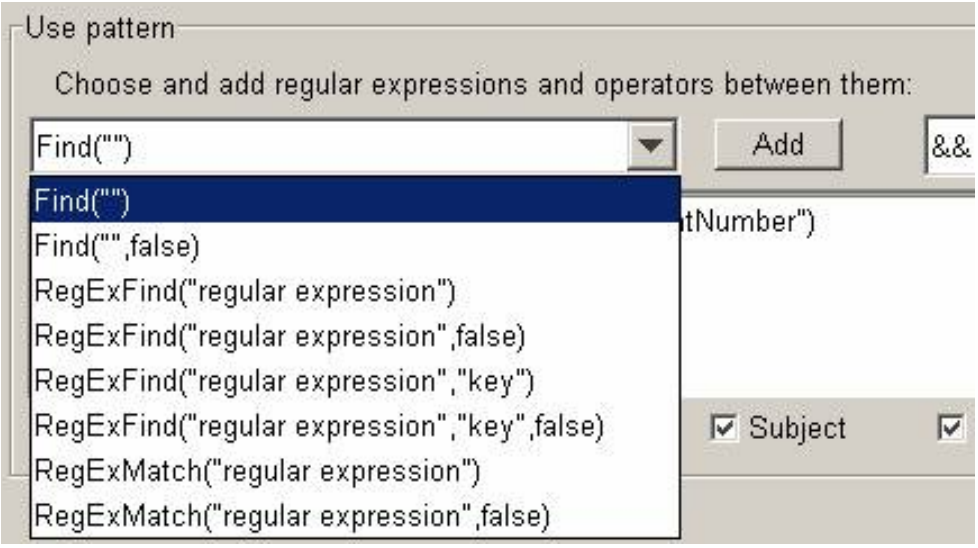


Figure 30: Regular Expressions in Knowledge Manager

After you select a form and click `Add`, you must put text between the quotation marks. More specifically, you must:

- For `Find`, put text between the empty quotation marks.
- For `RegExFind` and `RegExMatch`, substitute your desired text for `regular expression` and/or `key`.

Regular Expressions

A regular expression stands for, not one particular character string, but a class of character strings. For example, suppose that you want to find all interactions with U.S. Zip codes in them. U.S. Zip codes are five-digit numbers, so you could in theory write about 9,000 screening rules (`Find("00000")`, `Find("00001")`, `Find("00002")`, and so on). Fortunately, you can use the special symbol `\d`, which stands for any digit, to write a screening rule using a regular expression: `RegExFind("\d\d\d\d\d")`. This screening rule matches any sequence of five digits.

There are often several different ways of writing the same regular expression. For instance, two items separated by a hyphen and enclosed in square brackets denotes a range of which the two items are endpoints. So `[a-d]` matches a, b, c, or d, and `[5-8]` matches any digit between 5 and 8; hence `\d` is the same as `[0-9]`.

Note: In general usage, apart from Genesys Multimedia, regular expressions are case sensitive. However, in Knowledge Manager regular expressions are not case sensitive unless you add `,false` as described above.

Table 3 lists some of the most commonly-used elements of regular expressions:

Table 3: Elements of Regular Expressions

Symbol	Meaning	Example
.	Any character, including space	`b.t` matches *bat, bet, bit,* and *but.*
\d	Any digit	`\d\d` matches any pair of digits from 00 to 99.
\s	Space	`\d\s\d` matches 1 0, 5 9, and so on.
*	Zero or more instances of the preceding expression	`o*f` matches *oof, of,* and *f.* `me.*d` matches *med, mead,* and *meed.*
+	One or more instances of the preceding expression	`bre+d` matches *bred, breed* and *breeed.*
?	Zero or one instances of the preceding expression	`c?rude` matches *rude* and *crude.*
{x}	X instances of the preceding expression	`st.{2}k` matches *steak, stork*, and *stink.*
^	Any character except the following	`s[^e]t` matches *sat, sit,* and *sot,* but not *set.*
[ ]	Any characters or ranges within the brackets	Any characters: `b[aeiou]at` matches *boat* but not *brat.* Any range(s): `[0-9]th` matches *5th, 6th, 7th* `[a-z]` matches any lowercase letter; `[A-Z]` matches any uppercase letter.
\	Turns off the special meaning of the following symbol	`\*` matches the character * (asterisk); `\.` matches the character . (period or full stop).
|	Or	`[b|p]ig` matches *big* and *pig.* Do not be confused: | means *or* in regular expressions, but || means *or* as one of the Operators used in screening rule formulas.

Here are some other points to keep in mind:

- Space is just another character. The regular expression `savings account` contains a space, and so it does not match the string *savingsaccount.*
- Word boundaries are not considered. The regular expression `read` matches not only *read,* but also *reader, ready, spread, bread,* and so on.
- Use parentheses to group parts of regular expressions together. For example, `RegExFind("(\d{3}\.){2}")` puts `\d{3}\.` in parentheses so that the number-of-instances item `{2}` applies to the all of `d{3}\.`, not just to `\.`. This expression matches any group of three digits plus period plus any three digits plus period (for example, 198.351.). Further examples are provided in "Examples" on page 75.
- Regular expressions make use of many more special characters and operators than those listed in Table 3. Much documentation on regular expressions is available on the Web. Because Genesys Knowledge Management uses Java classes for regular expressions, it is best to consult documents based on the particular version of regular expressions used in Java. For example, a good reference was recently available at `http://java.sun.com/j2se/1.4.2/docs/api/java/util/regex/Pattern.html.`

Examples

This section provides examples of screening rules.

Credit Card Number

To find text that includes a typical credit card number, you need to match a sequence of four groups of four digits, each group separated by `-` (hyphen):

```
\d\d\d\d\-\d\d\d\d\-\d\d\d\d\-\d\d\d\d
```

Note: This regular expression also works without the `\` (backslash) before the hyphens. However, it is better practice to write `\-` for the character hyphen, because the hyphen also has a special use in range expressions like `[a-z]`.

Or if you want to allow for the possibility that some people will omit the hyphens, use `?` to make the hyphen optional:

```
\d\d\d\d\-?\d\d\d\d\-?\d\d\d\d\-?\d\d\d\d
```

You could also use the repetition notation to shorten each `\d\d\d\d` to `\d{4}`.

North American Phone Number

North American phone numbers consists of ten digits, grouped into two groups of three and one of four. There are a number of ways for the groups to be separated:

203-555-1234

(203) 555-1234

(203)555-1234

203 555-1234

203.555.1234

The following regular expression matches all of the above:

```
(\d\d\d|\(\d\d\d\))[\s\.\-]?\s*\d\d\d[\-\.]\d\d\d\d
```

Table 4 analyzes this regular expression.

Table 4: Phone Number Regular Expression

Symbols	Meaning	Remarks
\d\d\d	Three digits	
\d\d\d|\(\d\d\d\)	Three digits, or three digits enclosed in parentheses	\ turns off the special meaning of the character (
[\s\.\-]?	Space or period or hyphen or zero	Any one of the items enclosed in square brackets, either once or not at all
\s*	Zero or more spaces	
\d\d\d	Three digits	
[\-\.]	Hyphen or period	Note again the need to use \
\d\d\d\d	Four digits	

Telltale Words

To screen for interactions from dissatisfied customers, you might try a regular expression like the following:

```
(not\s([a-z]+\s)*(pleased | satisfied)) | unhappy | complain
```

The first part of this expression matches *not* followed by zero or more words followed by *pleased* or *satisfied*; for example, *not very pleased, not satisfied, not at all satisfied* (but it also matches strings like *can not believe how pleased I am*). The rest matches the single words *unhappy* and *complain.*

Importing and Exporting

You can import and export categories, standard responses, field codes, screening rules, training objects, and models (training objects and models are restricted to Genesys Content Analyzer; see Chapter 2, “Genesys Knowledge Management: Content Analyzer,” on page 83 for more information). Use exported files as backups or to transfer objects between environments.

Note: At this time you cannot import or export FAQ objects.

Procedure:
Exporting Knowledge Manager objects

Start of procedure

1. On any tab, select `Export`. The `Export` dialog box opens, as shown in Figure 31.

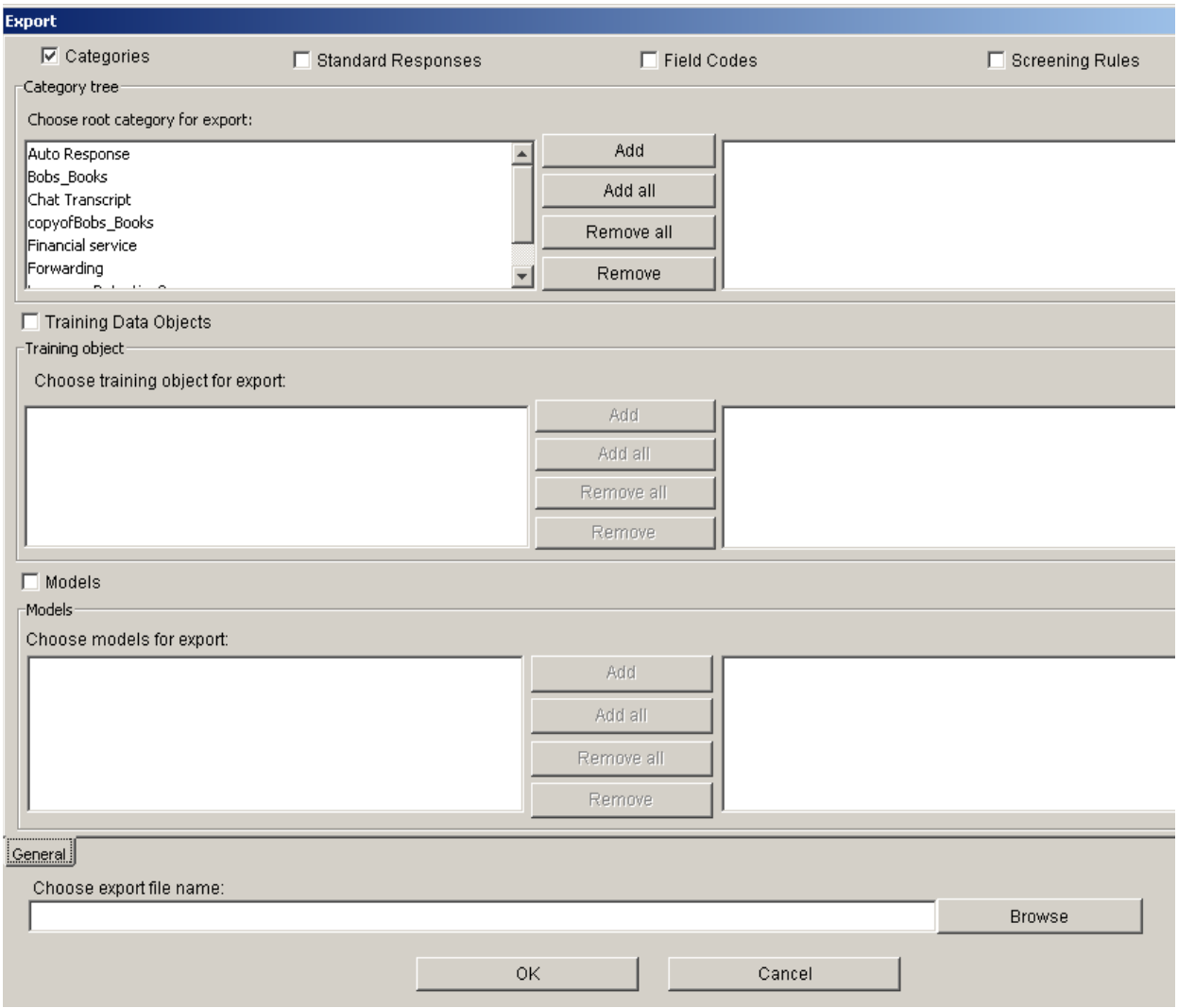


Figure 31: Export Dialog Box

2. Select the object(s) to export, using the check boxes at the top of the dialog box.
   - Categories—Exporting categories is a prerequisite for exporting standard responses, training objects, or models.
   - Standard Responses—Knowledge Manager exports the standard responses belonging to the category tree that you selected.
   - Field Codes—No selection is possible. You must export all or none.

- Screening Rules—If you select this checkbox, a Screening Rules tab appears towards the bottom of the dialog box. You must make further selections on this tab, as described in Step 4

3. Make further selections for category trees, training objects, and models.
   - Each type of object has two fields in the Export dialog box: one on the left listing available objects, another on the right listing the objects selected for export. Use the Add/Add all/Remove all/Remove buttons to move objects between the two fields.
   - The category structures that are selected (that appear in the right-hand category field) determine which training objects and models appear in the Training object and Models areas.
4. Make further selections for screening rules: On the `Screening Rules` tab, shown in Figure 32, move rules from the list of all rules (left-hand field) to the list of selected rules (right-hand field).

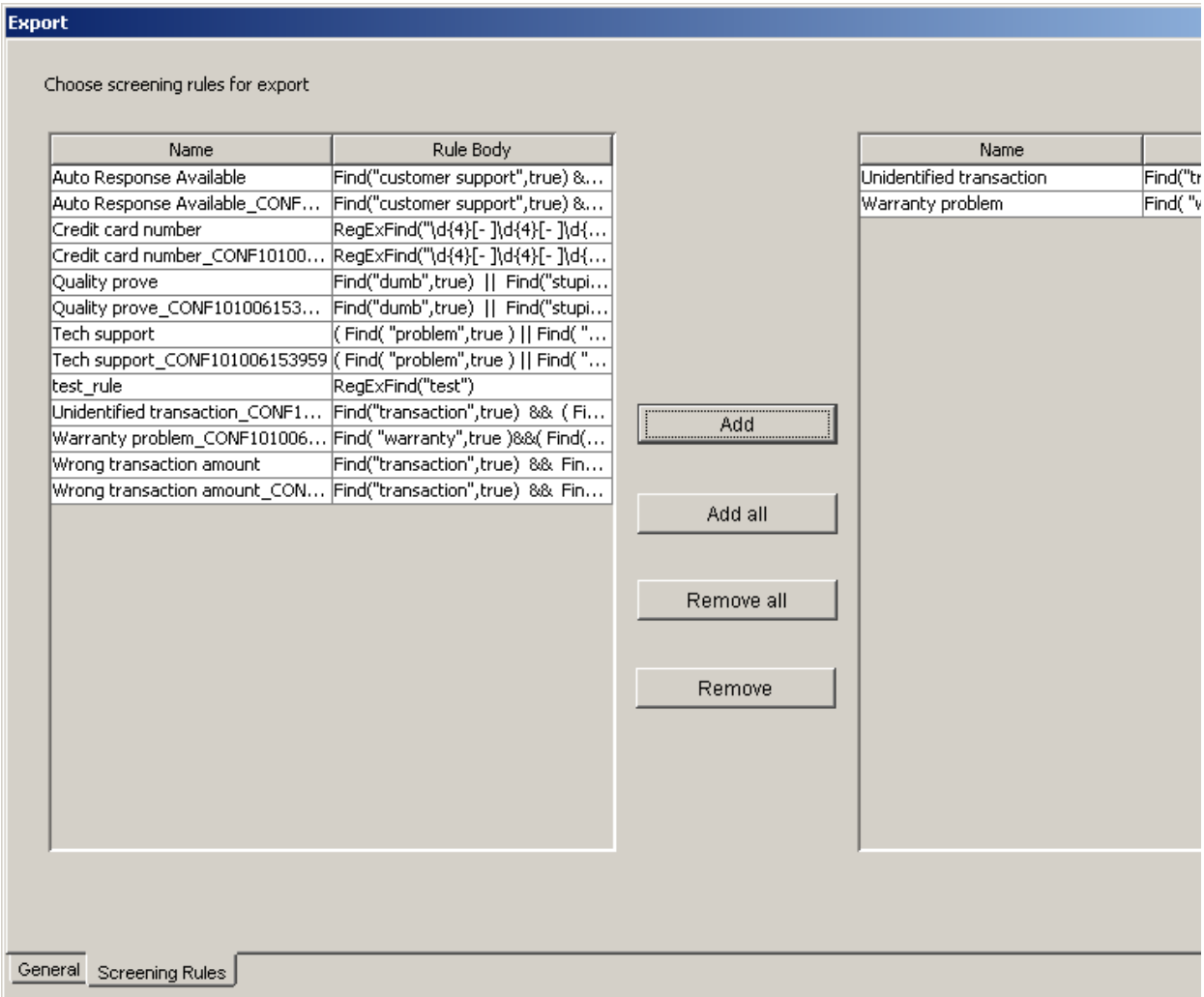


Figure 32: Export: Screening Rules Tab

Note: When selecting objects to export, keep in mind that importing will import all objects in the exported file. You cannot select which objects to import.

5. Enter a name for the exported file and click `OK`. The file is created in the directory containing the Knowledge Manager executable.
6. A message appears reporting on any errors or warnings encountered (see Figure 33).

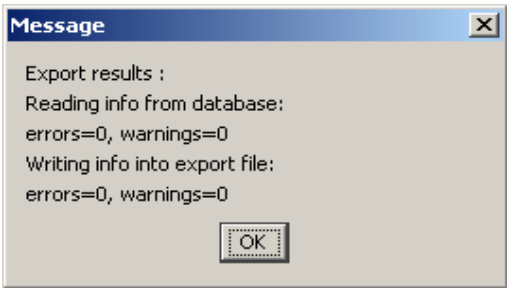


Figure 33: Export Results Message

End of procedure

Procedure: Importing Knowledge Manager objects

Start of procedure

1. On any tab, select `Import`. The `Import` dialog box opens, as in Figure 34.

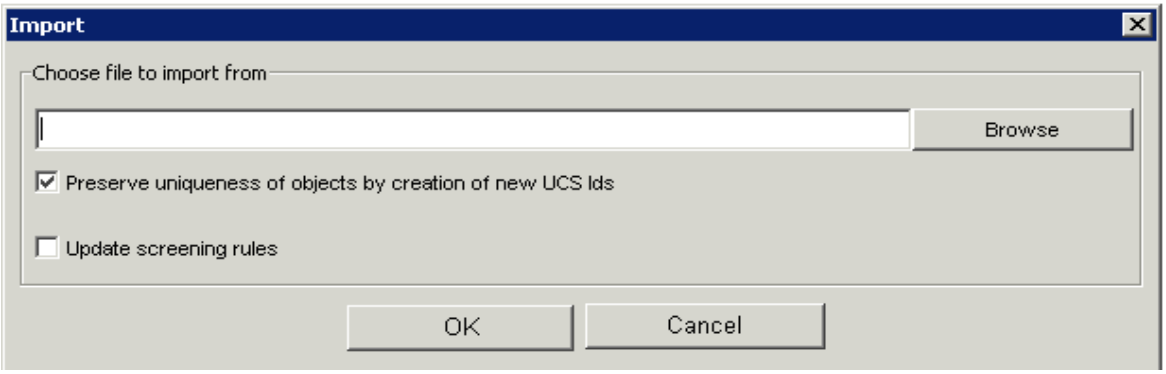


Figure 34: Import Dialog Box

2. Browse to the file that you want to import, or enter its path and name.

   The check box `Preserve uniqueness of objects by creation of new UCS Ids` controls whether the imported records receive new database IDs. Genesys strongly recommends that you leave this check box selected; otherwise, the imported records keep their old IDs and there is a risk of

creating uniqueness conflicts. It is only safe to keep old IDs when you are importing into an empty database. One reason to keep the old IDs would be to preserve compatibility with non–Knowledge Manager components (such as a routing strategy) that need to refer to them.

Note: If the `Preserve uniqueness` check box is cleared and an imported object comes in with an ID that is identical with an existing object's ID, the import process cancels.

The check box `Update screening rules` controls whether imported screening rules overwrite existing screening rules with the same name. If this check box is not selected, screening rules are treated like all other Knowledge Management objects, as described in the preceding paragraph. If the check box is selected, screening rules are treated differently from all other objects: If the names match, the imported screening rule replaces the existing one.

Note: If the imported rule's name does not match any existing rule but its database ID happens to match that of an existing rule, then Knowledge Manager creates a new ID for the imported rule.

3. If a root category being imported has the same name as an existing category, Knowledge Manager asks you to change the name of the category being imported.

   If other objects have the same name as existing objects, Knowledge Manager appends `_<hms>` to the name of the imported object. `<hms>` is a timestamp where `h` is the hour (using a 12-hour clock), `m` is the minutes, and `s` is the seconds. Each unit may be one or two digits; there is no padding. For example, if at 4:25:07 PM you import a screening rule called `Sales`, and there is also an existing rule called `Sales`, the new name of the imported rule is `Sales_4257`.

   Note: This adds between four and seven characters to the name of the object. You should be especially careful of this if any imported object's name is more than 58 characters long: the added characters may produce a new name that violates the 64-character limit on names of Knowledge Management objects (see also page 26). Importing may fail on objects with names that are too long.

4. Click `OK`. All objects in the selected file are imported.
5. A message appears reporting on any errors or warnings encountered (see Figure 35).

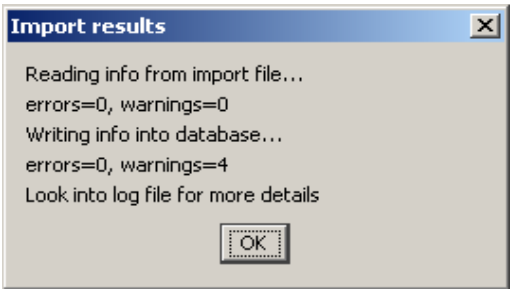


Figure 35: Import Results Message

End of procedure

Typical Response Times

This section describes some typical response times for Knowledge Manager tasks. These figures are for a machine with two Pentium 4 processors and 1 GB of RAM.

- Loading and refreshing a category tree takes 4–6 seconds per 2000 categories (each category having one standard response).
- Copying and pasting a category tree takes approximately 100 seconds per 1000 categories.
- Deleting a category tree takes approximately 100 seconds per 1000 categories.

Chapter

2 Genesys Knowledge Management: Content Analyzer

This chapter describes using the Genesys Content Analyzer option for Knowledge Manager. It covers these topics:

- Overview, page 83
- Training, page 84
- Testing Models, page 119
- Using and Rating Models, page 128
- Design and Use Considerations, page 138
- Language Detection Model, page 141
- FAQ Objects, page 142
- Typical Response Times, page 156

Overview

Genesys Content Analyzer is an option to Genesys E-mail, requiring an additional license. It adds natural language processing technology to Genesys Knowledge Management.

See the "Overview" section in Chapter 1, "Genesys Knowledge Management: Basics," on page 19 for an outline of the structure of the Knowledge Manager user interface. Many of the terms and concepts relevant to Genesys Content Analyzer are also defined in the glossary, which begins on page 241.

Models

Genesys Content Analyzer applies a classification model—a statistical representation of a category tree—to an incoming interaction and produces a list of the categories that the interaction is most likely to belong to. Each likely

category is assigned a percentage rating indicating the probability that the interaction belongs to this category.

Note: Classification, like screening (see "Screening Rules" on page 58) can operate on any interaction that has text somehow associated with it, whether as the body of the interaction (e-mail, chat), or otherwise (as user data, for example). In practice, it is expected that most interactions that are screened or classified will be e-mail messages; therefore this Guide uses the terms *e-mail* and *message* interchangeably to refer to these interactions. In fact whatever is said here about e-mail applies to any interaction that has associated text.

Training Objects

The process of creating a model is called *training.* Training operates on a *training object,* which is a category tree plus a set of text objects, with each text object assigned to one category in the tree.

Import and Export

You can import and export training objects and models. This is also a means of copying training objects. See "Importing and Exporting" on page 77 of the previous chapter.

Components

Genesys Content Analyzer does not have components as such. Rather, it adds functionality to the components of Genesys Knowledge Management:

- It activates Training Server, which has no function in the basic Genesys Knowledge Management but is required for training models.
- It enables Classification Server to categorize incoming interactions using models.
- It enables Knowledge Manager to control the creation of training objects, classification models, and FAQ objects.

Training

Training consists of the following steps, described in this section:

1. Create a training object.
2. Schedule and run the training.

All of these actions are performed by Training Server according to settings you provide using Knowledge Manager, which also displays the results.

Creating a Training Object

A training object combines a category tree and a set of text objects, with each text object assigned to one category in the tree (categorized). The text objects are typically e-mails, but you may choose to have the set of text objects also include the standard responses associated with the category tree.

Training scans the text objects and forms a statistical model of the words and phrases that tend to occur in each category. This is why you may want to have the training scan standard responses: they are very likely to include many words and phrases diagnostic of their category. This is also the reason that more text objects is better: it increases the sample size, which increases the accuracy of the model.

There are five possible sources of categorized text objects:

- E-mails that agents have assigned to categories, as described in this section
- Text objects (in the form of e-mails) that you create in Knowledge Manager, as described in "Creating new e-mails manually" on page 91
- Standard responses, as described in "Adding standard responses or other training objects" on page 93
- Other training objects, as described in "Adding standard responses or other training objects" on page 93
- Uncategorized e-mails that you categorize using the TO Data Analyzer, as described in "Adding Uncategorized E-Mails Using the TO Data Analyzer" on page 95

Once you have a category tree or trees (see "Creating a Category Tree" on page 25), you can create a training object.

Procedure: Creating a new training object

Purpose: To create an object that can be used to produce a classification model.

Prerequisites

- A category tree

Start of procedure

1. In Knowledge Manager, select a tenant and language. This determines which category trees are available. See "Notes on Language" on page 115 for some considerations relevant to choosing a language.
2. On the Training tab, select `File > New`, click the `New` icon, or right-click anywhere and select `New Training Object` on the shortcut menu. The `New Training Object` dialog box appears, as shown in Figure 36.

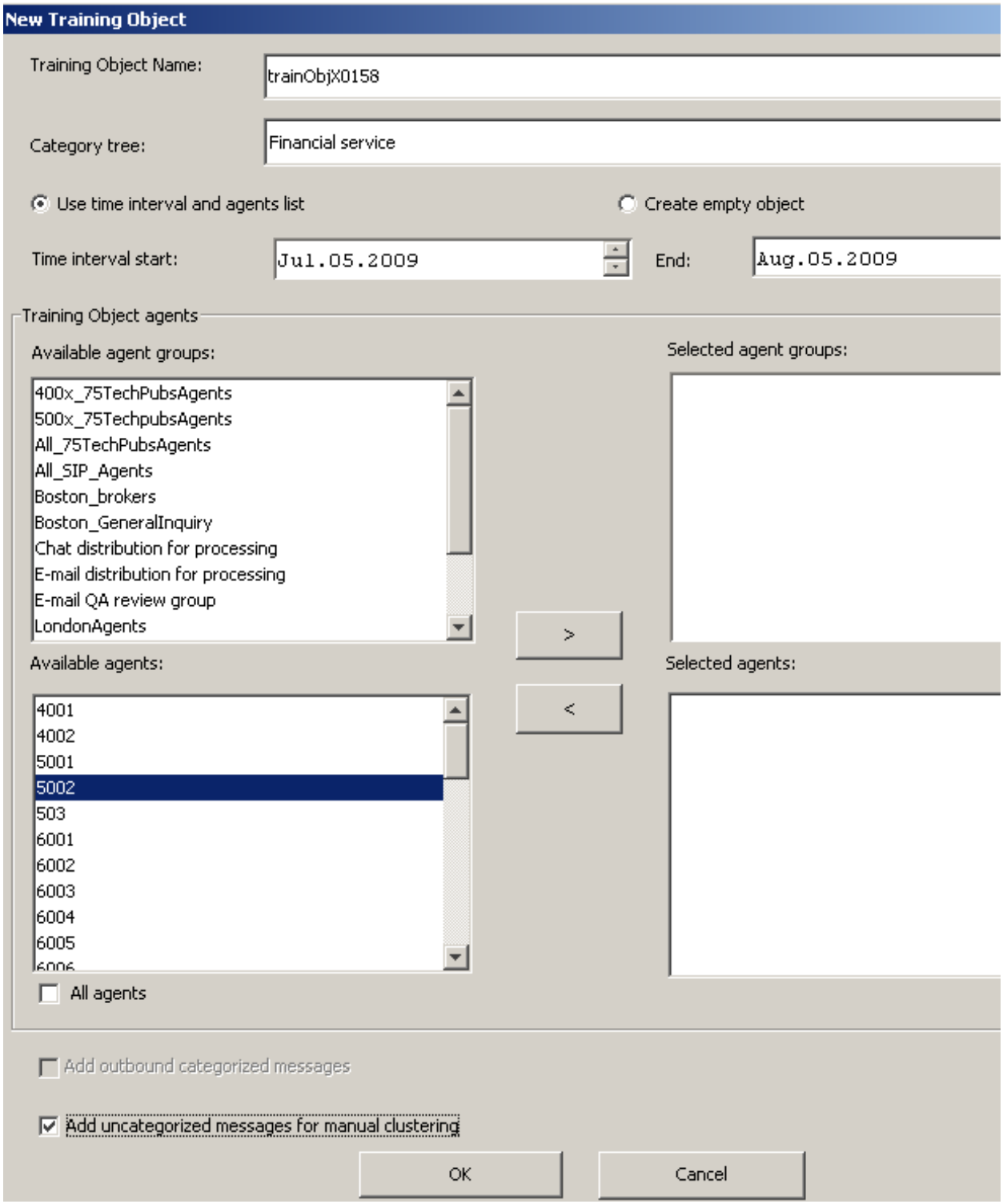


Figure 36: New Training Object Dialog Box

3. Enter a name for the object (see page 26 in Chapter 2 for restrictions on the names of Knowledge Manager objects) and select a category structure (category tree).

4. You now have two alternatives, depending on whether you already have a collection of e-mails that have been categorized:
   a. If you do not already have a collection of categorized e-mails, select the `Create empty object` radio button.
   b. If you already have categorized e-mails, the categories were assigned to each e-mail by the agent who handled the interaction. In adding categorized e-mails to a training object, you can simply add all e-mails categorized by any agent at any time, or you can make the following adjustments:
      - Use only the e-mails that were categorized during a certain time period. You may want to exclude earlier times when agents were still learning to use the category system. To do this, set dates in the `Time interval start` and `End` fields.
      - Use only the e-mails that were categorized by selected agents or groups (some agents or groups may be especially reliable in their choice of categories). To do this, use the > and < buttons to move agent names from the `Available agent groups` window or `Available agents` window to the `Selected agent groups` window or the `Selected agents` window.

Notes: Agent names appear in alphabetical order.

For agent groups to appear in the `Available agent groups` box, they must be configured in the Tenant that you are creating this training object in.

To move individual agents, you must first clear the `All agents` check box

5. The `Add outbound categorized messages` check box is enabled only if Knowledge Manager is operating with UCS 7.5. See further explanation below.
6. Select the `Add uncategorized messages for manual clustering` check box if you want to add uncategorized e-mails to the root category. You can then use the TO Data Analyzer to categorize them, as described in "Adding Uncategorized E-Mails Using the TO Data Analyzer" on page 95.

Note: The uncategorized e-mails added by this step are from the time interval and/or agents that you designated in the previous step. If you made no designations, the uncategorized e-mails are from your entire UCS database.

End of procedure

Next Steps

- If you are operating Knowledge Manager with UCS 7.5, review "Add Outbound Messages" immediately below.
- To add more text objects, continue with "Adding More Text Objects to the Training Object" on page 91 and "Adding Uncategorized E-Mails Using the TO Data Analyzer" on page 95.
- To proceed with training, continue with "Schedule Training" on page 108.

Add Outbound Messages

The `Add outbound categorized messages` check box, shown in Figure 36 on page 86, behaves differently depending on the version of UCS that Knowledge Manager is operating with.

- If Knowledge Manager is operating with UCS 7.6, the checkbox is disabled. The new training object includes both of the following (as delineated by any time interval and agent list that you set):
  - Categorized inbound e-mails
  - Uncategorized inbound e-mails whose parent is a categorized outbound e-mail.
- If Knowledge Manager is operating with UCS 7.5, the check box is enabled, and has the following effects:
  - Not selected (the default): The training object includes only categorized messages that are inbound and are first in their thread.
  - Selected: The training object includes all categorized messages, both inbound and outbound.

Selecting this check box may be helpful in a situation in which agents do not categorize inbound e-mails, but do categorize the outbound e-mails that they generate in reply. In such a situation you may have an unacceptably small number of categorized e-mails unless you add categorized outbound e-mails. However, note that selecting this check box adds all outbound categorized messages, and those outbound messages that are replies may have content that is not very relevant to the category of the original e-mail.

Copying a Training Object

There are two ways to copy a training object: as an exact copy, or with selected text filtered out.

Procedure: Making an exact copy of a training object

Summary

To make an exact copy of a training object, first export the training object, then import it.

Start of procedure

1. In the Export dialog box:
   a. Select the category tree.
   b. Select the training object.
2. When importing, select new database IDs (see "Importing Knowledge Manager objects" on page 80). Knowledge Manager asks you to modify the name of the category tree being imported.

End of procedure

Procedure: Making a filtered copy of a training object

Summary

To make a copy that filters out text that you specify, proceed as follows.

Start of procedure

1. On the left-hand pane of the `Training` tab, select the root `Training Objects` node.
2. On the right-hand pane, highlight and right-click the training object that you want to make a filtered copy of.
3. Select `Create filtered copy of Training Object` from the resulting drop-down list.

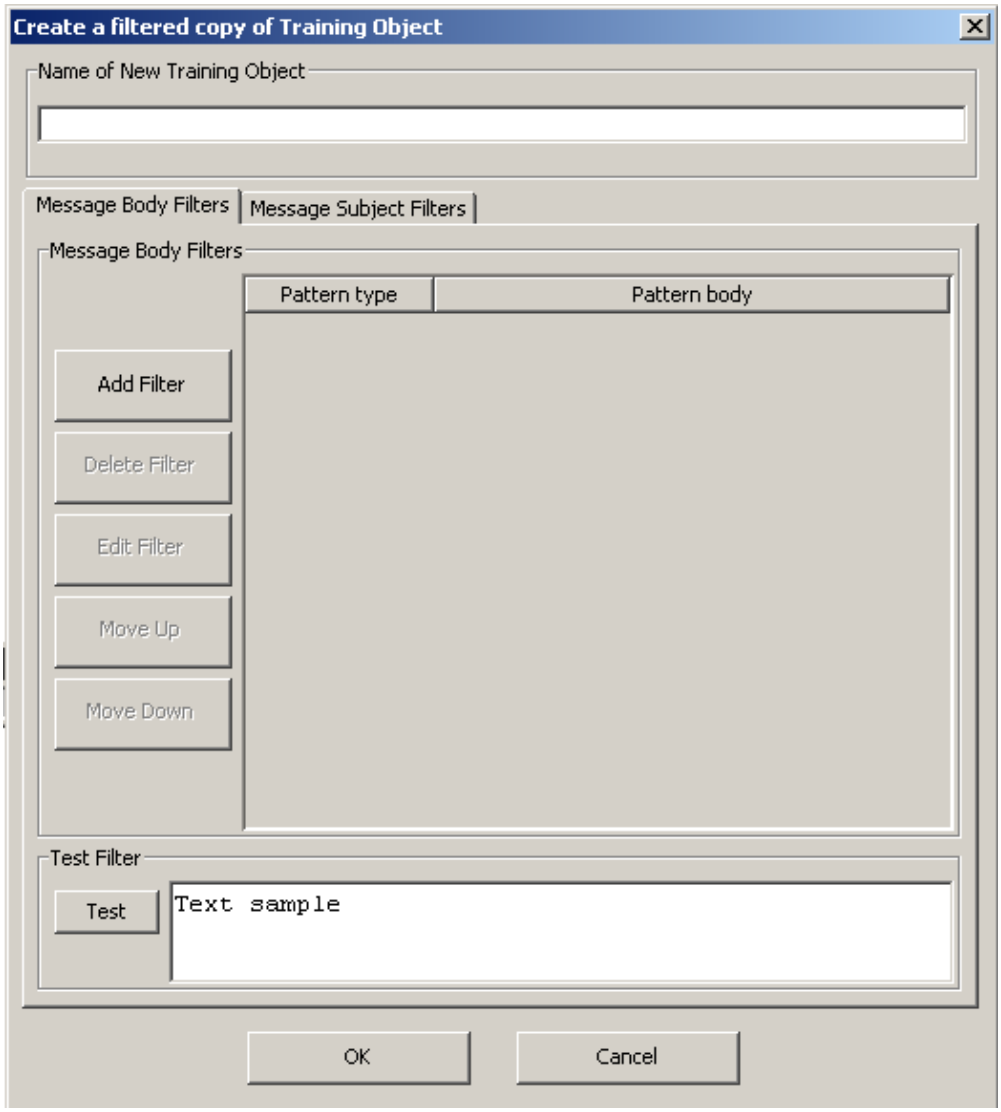


Figure 37: Create a Filtered Copy of Training Object

4. On the resulting `Create filtered copy of Training Object` dialog box, shown in Figure 37, proceed as follows:
   a. Enter a name for the new filtered copy of the training object.
   b. Create filters, and test them if you wish.

   This works identically with the `Text Preprocessing` tab of the `Model Training Schedule` dialog box, which is described on page 111, except that here you create separate filters for the body and subject of messages, using the `Message Body Filters` and `Message Subject Filters` tabs.
   c. Click OK to save the new filtered copy of the training object.

End of procedure

Adding More Text Objects to the Training Object

Regardless of whether you have previously-categorized e-mails that you included in your training object, you can add more text objects to it in the following ways:

- Create new e-mails manually, as described on page 91.
- Add standard responses or other training objects, as described on page 93.
- Add uncategorized e-mails using the TO Data Analyzer utility, as described on page 95.

Procedure: Creating new e-mails manually

Purpose: To increase the accuracy of a model produced by a training object by adding new manually-created e-mails to the training object.

Start of procedure

1. Close the `New Training Object` dialog box. On the left-hand pane of the `Training` tab, select the training object. This produces a three-pane view, shown in Figure 38.

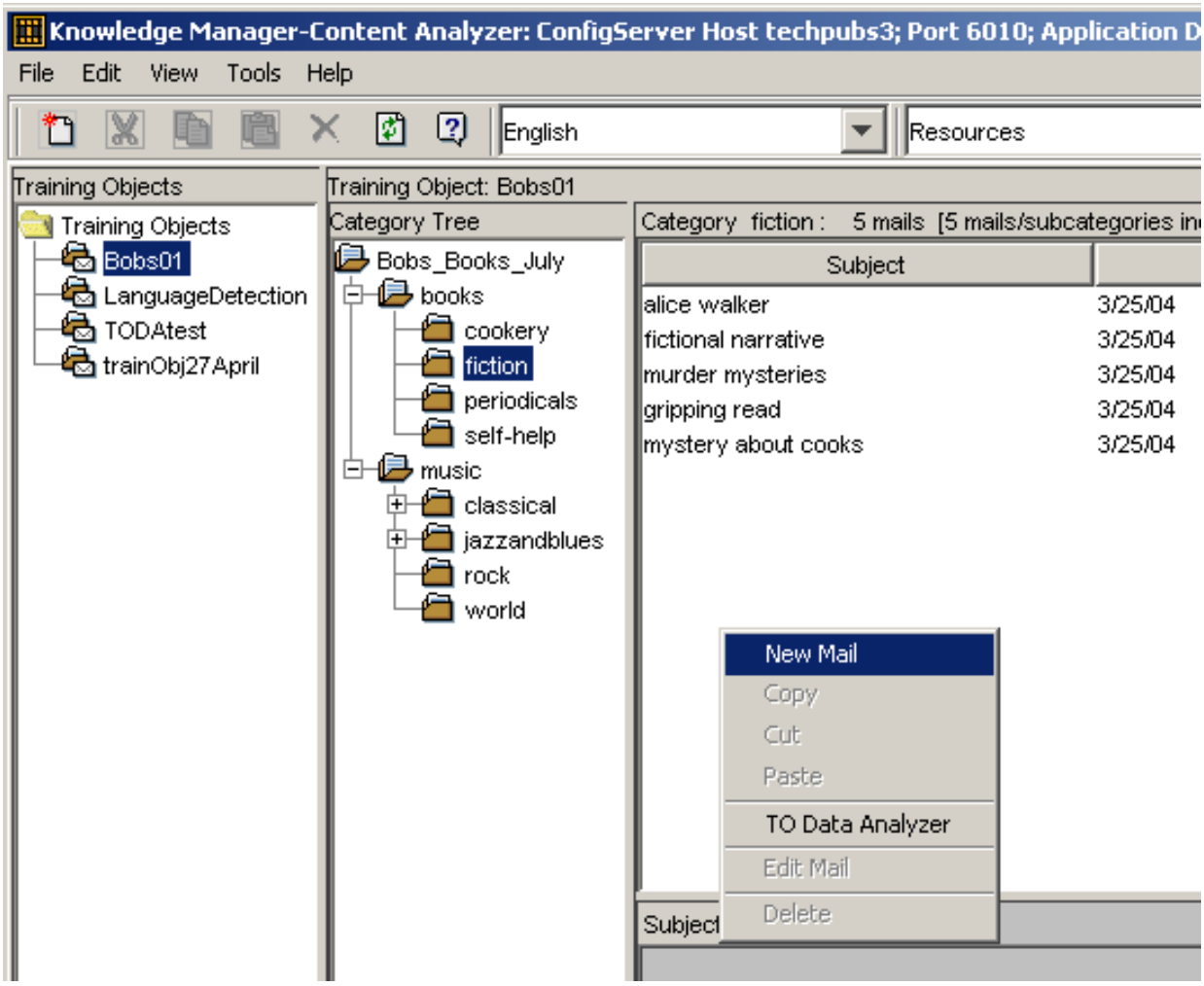


Figure 38: Add E-mail to a Training Object

2. On the center pane, select the category that you want to add an e-mail to. Figure 38 shows a user about to add a new e-mail to the `fiction` category.
3. On the right-hand pane, right-click and select `New Mail` from the shortcut menu. The `Mail Editor` opens, as shown in Figure 39.

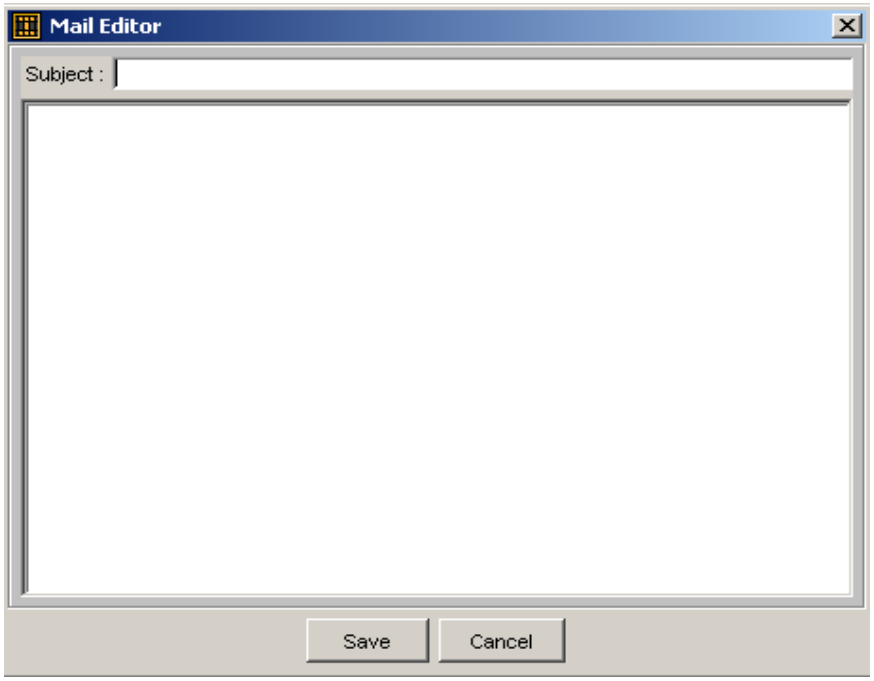


Figure 39: Mail Editor

4. Enter text into the `Subject` and `Body` fields, then click `Save`.

   E-mails that you add in this way are stored as being handled by the agent called `default`.

End of procedure

Procedure: Adding standard responses or other training objects

Purpose: To increase the accuracy of a model produced by a training object by adding either standard responses or other training objects to the training object.

Start of procedure

1. To add standard responses, go to the left-hand pane of the `Training` tab and select the root `Training Objects` node. This produces a two-pane view, as shown in Figure 40 on page 94.

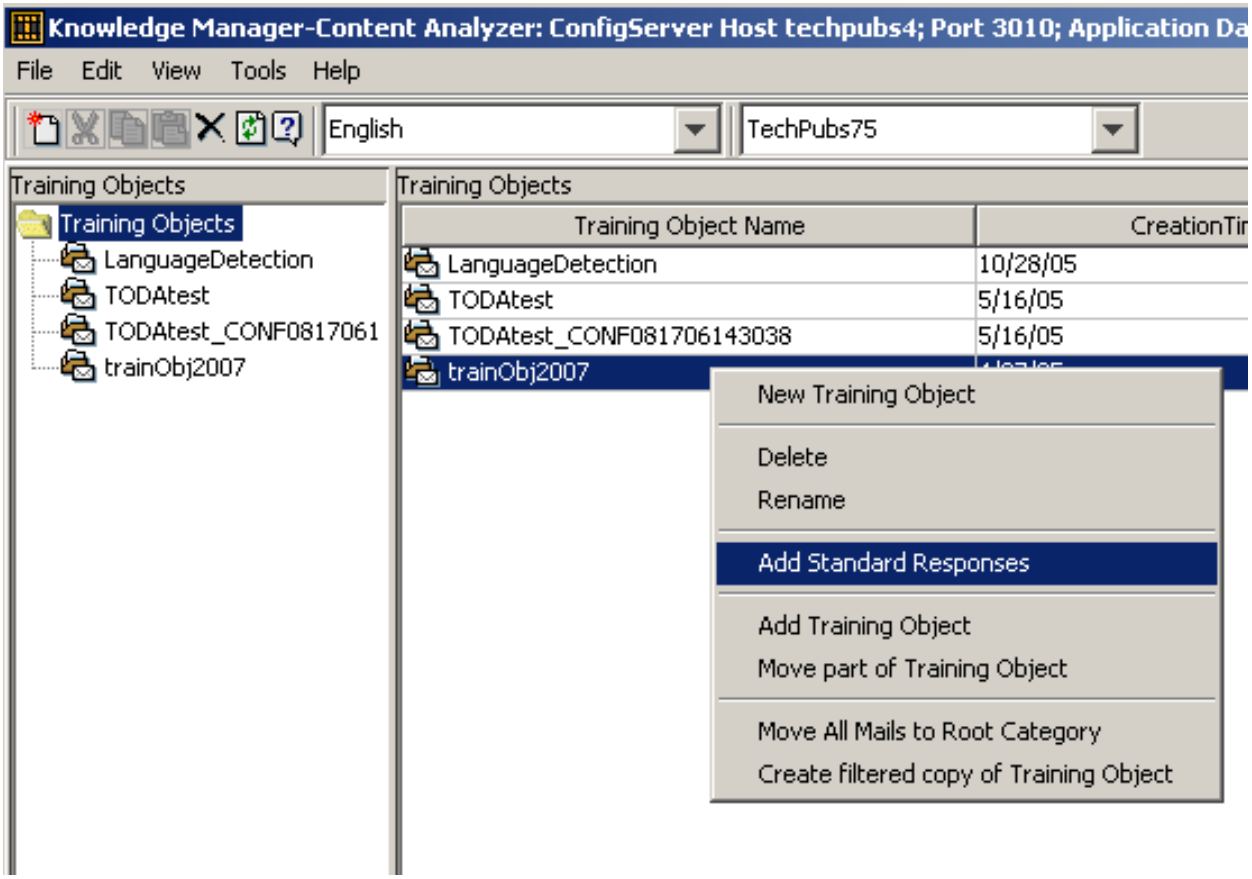


Figure 40: Add Standard Responses to a Training Object

2. Right-click a training object on the right-hand pane, then choose `Add Standard Responses` from the shortcut menu. Figure 40 shows a user about to add standard responses to the `trainObj2007` training object.
3. To add a training object, repeat Steps 1 and 2, but choose `Add Training Object` from the shortcut menu.

Note: You can add a training object to another training object only if both training objects use the same category tree.

End of procedure

Adding Uncategorized E-Mails Using the TO Data Analyzer

If you have a sizeable database of uncategorized e-mails, the TO (Training Object) Data Analyzer can help you to:

- Group the uncategorized e-mails and build a category tree for them.
- Assign the uncategorized e-mails to existing categories.

In the most basic terms, the TO Data Analyzer does the following:

1. It shows you all uncategorized e-mails one at a time, in an order determined by criteria that you set (if you set no criteria, it shows them in the order in which they were created).
2. As each e-mail displays, you choose whether to include it in a holding area.
3. You then assign the e-mails in the holding area to a category. This can be an existing category or a new one that you create.

The TO Data Analyzer has the tabs listed in Table 5.

Table 5: TO Data Analyzer Tabs

Tab	What it Does	Location of Description
Main	Displays uncategorized e-mails one at a time for you to categorize	"Adding e-mails to a category on the Main tab" on this page
Search Criteria	Sets the criterion determining the order that uncategorized e-mails are displayed in	"Search Criteria Tab" on page 103
Indexing	Displays information on cooccurrence of words in uncategorized e-mails	"Indexing Tab" on page 104

Procedure: Adding e-mails to a category on the Main tab

Purpose: To use the TO Data Analyzer to search for uncategorized e-mails that are a likely fit for a selected category.

Prerequisites

For uncategorized e-mails to be available to the TO Data Analyzer, they must be associated with the root category of the category tree in the training object. You perform this association in either of the following ways:

- When you create the training object, select the `Add uncategorized messages for manual clustering` check box as described in Step 6 of "Creating a Training Object" on page 87.

- In an existing training object, you can move all e-mails to the root category (before you do this it is advisable to make a backup copy of the training object; see "Copying a Training Object" on page 88):
  a. On the left-hand pane, select the root `Training Objects` node to give the two-pane view.
  b. On the right-hand pane, right-click a training object and select `Move All Mails to Root Category`, as shown in Figure 41.

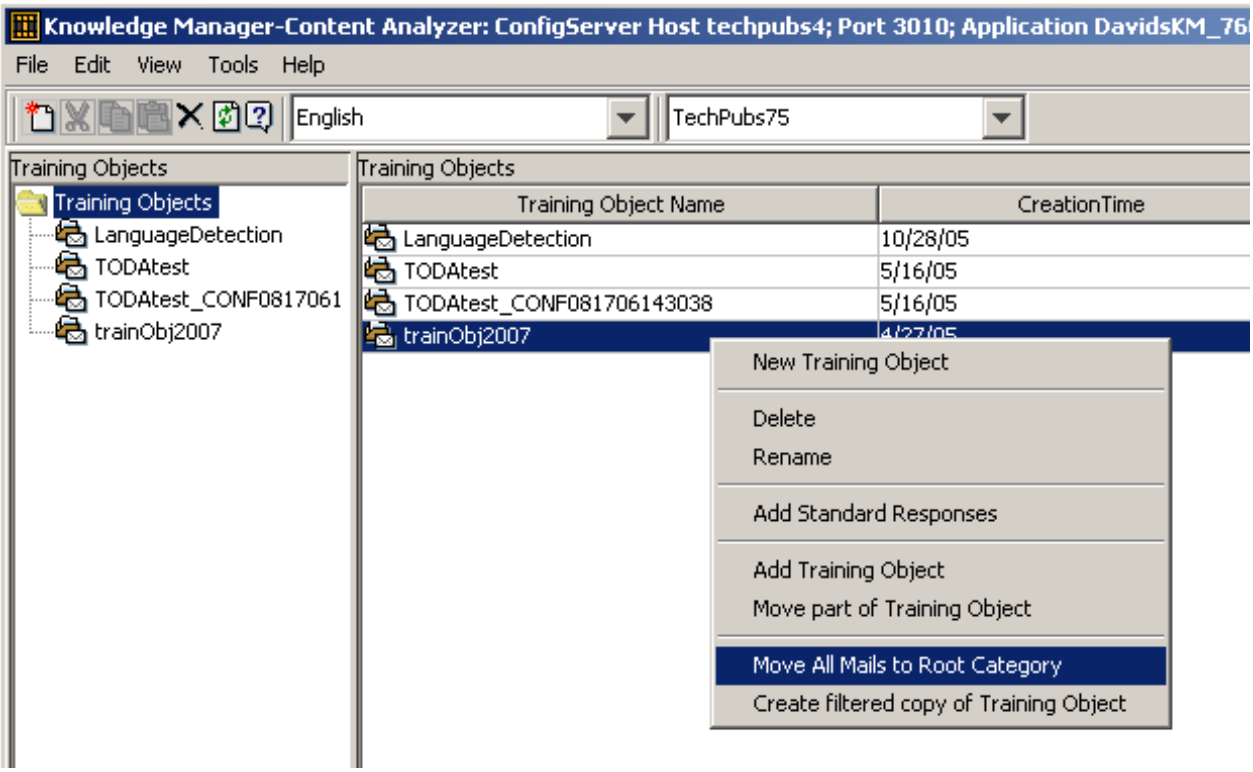


Figure 41: Move All E-Mails to the Root Category

Start of procedure

1. On the `Training` tab, select a training object.
2. On the middle pane, select a category that you want to add e-mails to.
3. On the rightmost pane, select the e-mails already assigned to this category.
4. Still on the rightmost pane, right-click and select `TO Data Analyzer`, as shown in Figure 42.

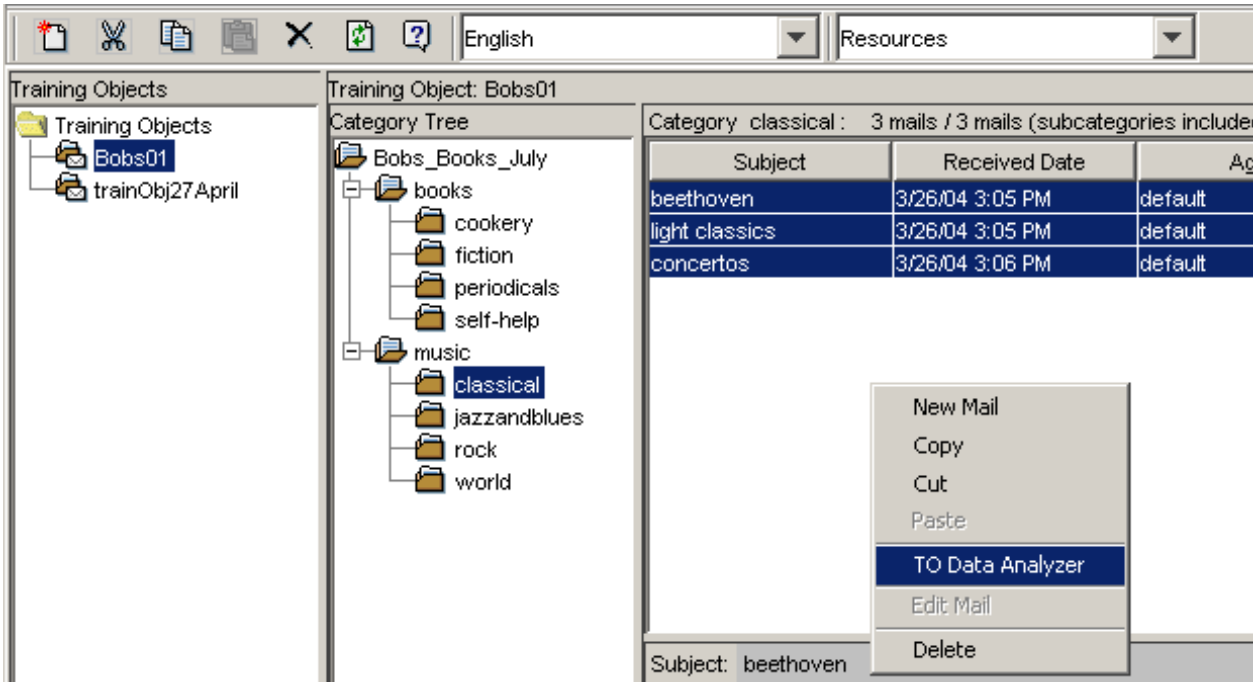


Figure 42: Start the Training Object Data Analyzer

5. The `Uncategorized Message Clustering` dialog box appears with its `Main` tab showing, as shown in Figure 43.

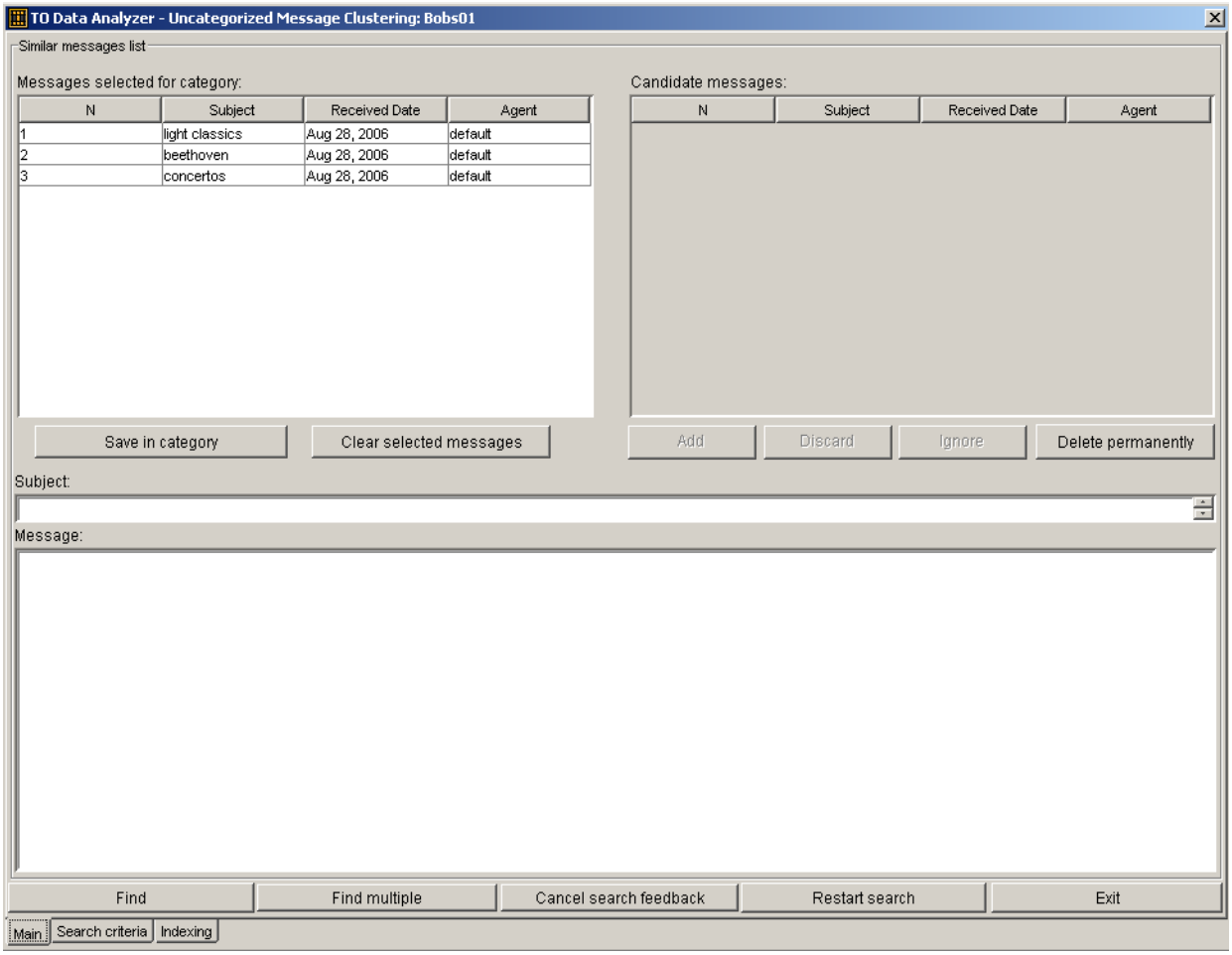


Figure 43: Uncategorized Message Clustering

Note that the e-mails that you selected in Step 3 appear in the `Similar messages list`. These e-mails are now part of the criteria that TO Data Analyzer uses to choose e-mails to display.

6. Click `Find`. The system displays a candidate e-mail as shown in Figure 44:
   - The `Candidate messages` box shows information about the e-mail. `N` is number, relevant only for the `Find multiple` command, described below.
   - The `Subject` box shows the subject of the e-mail.
   - The `Message` box shows the e-mail body.

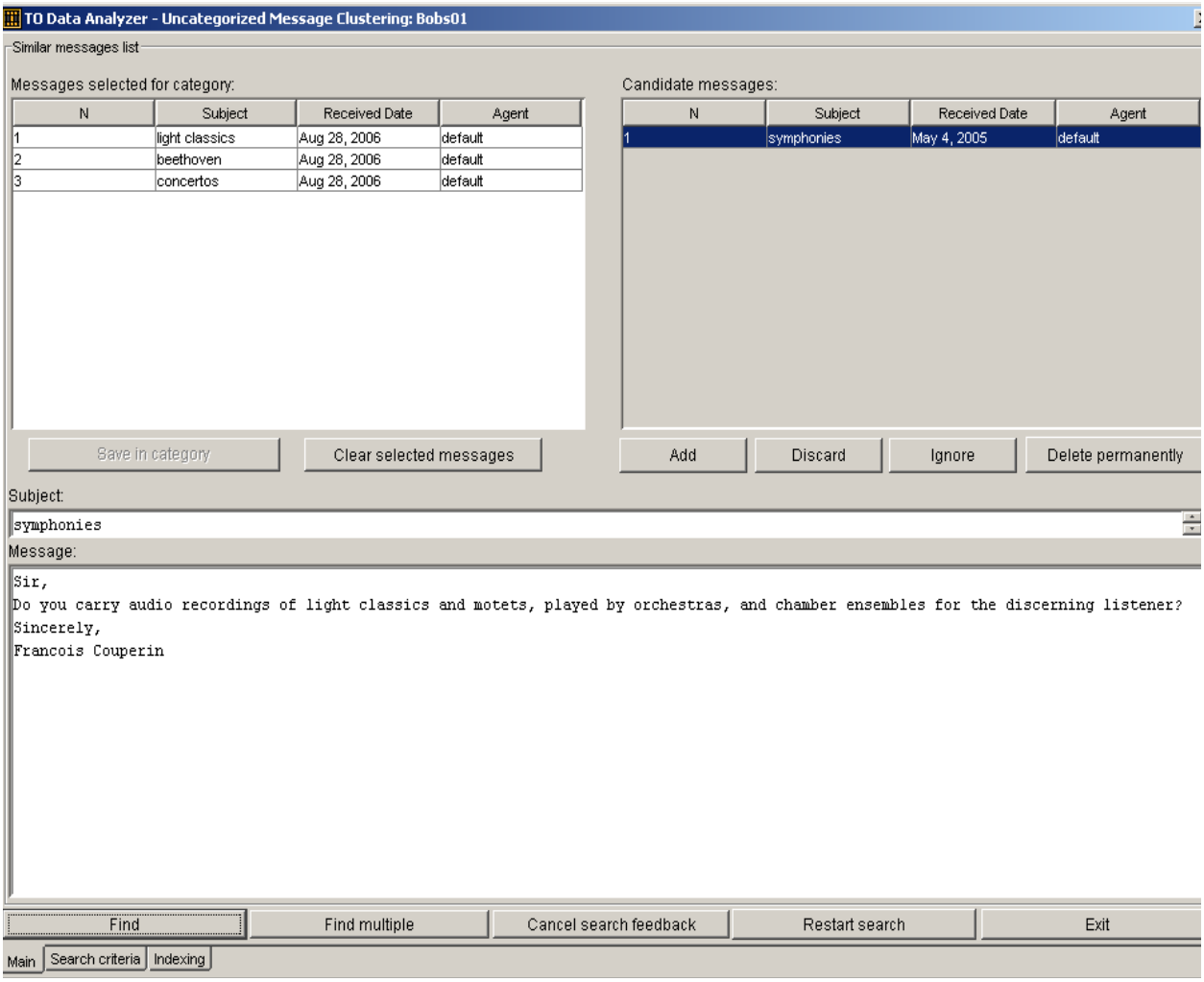


Figure 44: Message Clustering: First Candidate

As an alternative to `Find`, click `Find multiple`. The system displays the 15 e-mails that are the best candidates for the category, as shown in Figure 45.

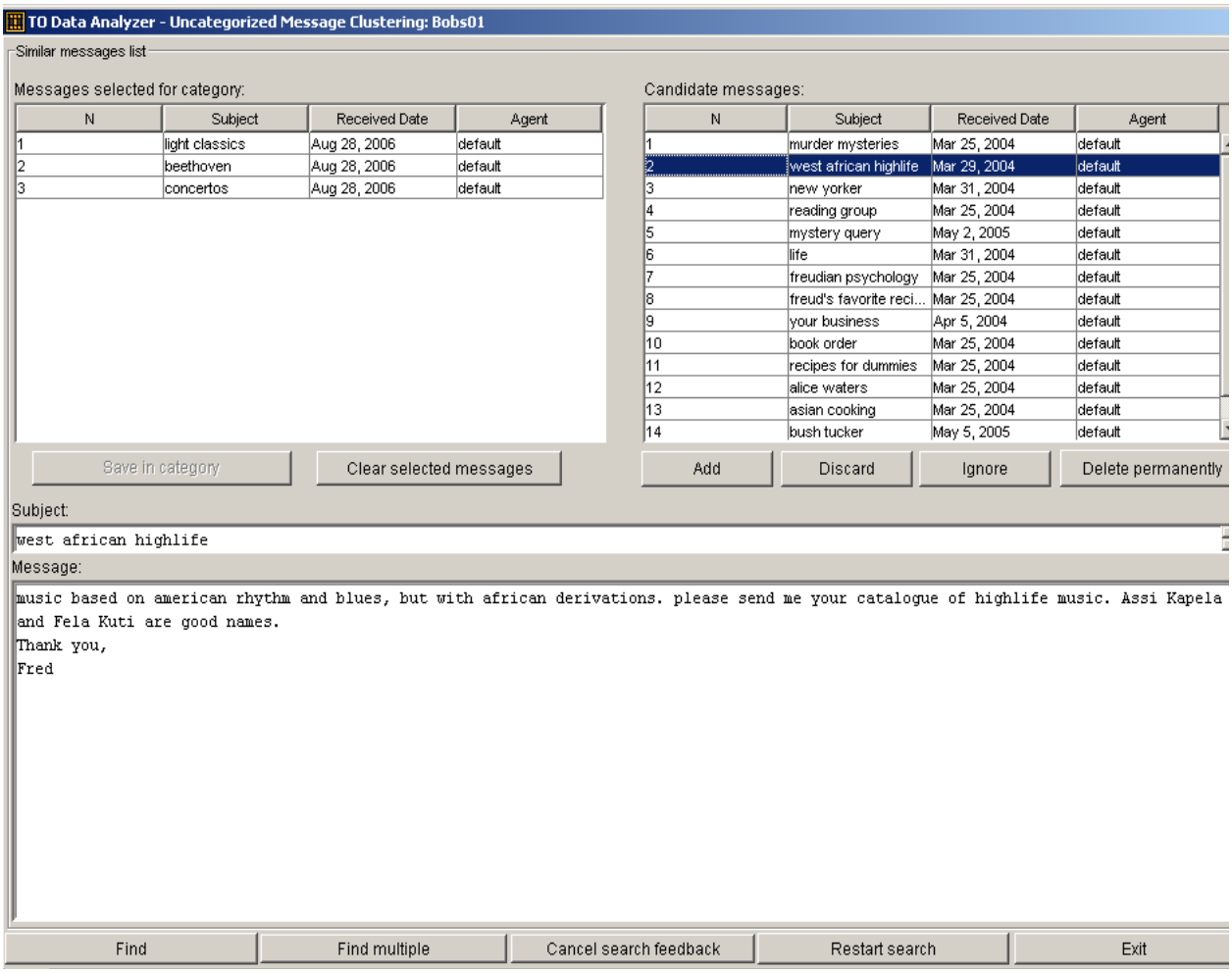


Figure 45: Message Clustering: Multiple Candidates

7. Use the buttons at the bottom of the dialog to select one of the following actions for this candidate (or candidates, if you clicked `Find multiple` and then used SHIFT-click or CTL-click to select multiple candidates):
   - `Add`—Add to the selected category and adjust the search accordingly (see "Search Criteria Tab" on page 103 for more explanation).
   - `Discard`—Omit from the selected category and adjust the search accordingly.
   - `Ignore`—Omit from the selected category but do not adjust the search.
   - `Delete permanently`—Remove from the training object permanently. This e-mail will not appear again.

Note: `Delete permanently` is useful for deleting garbage e-mails, that is, e-mails that you do not want to use anywhere in the training object. With `Discard` and `Ignore`, in contrast, the e-mail is no longer considered in the current search for the selected category, but it remains in the training object.

8. Click `Find` again, and the next candidate e-mail displays in the `Subject` and `Message` boxes. Decide whether to add, discard, or ignore this e-mail.
9. Continue in the same way, repeating Step 8. Knowledge Manager continues to display e-mails in the order determined by the criteria that you have set (see the next section, "Search Criteria Tab", for details on setting criteria). When there are no more uncategorized e-mails, `{ Mail not found }` displays in the `Message` box.
10. When you are done adding e-mails to the `Similar messages list`, click one of the following:
    - `Save in category`—This adds the e-mails to a category, as explained in the next step.
    - `Clear selected messages`—This clears all e-mails from the `Similar messages list`. You can start again by clicking `Find`.
11. If you click `Save in category`, the `Choose category` dialog box appears, as shown in Figure 46.

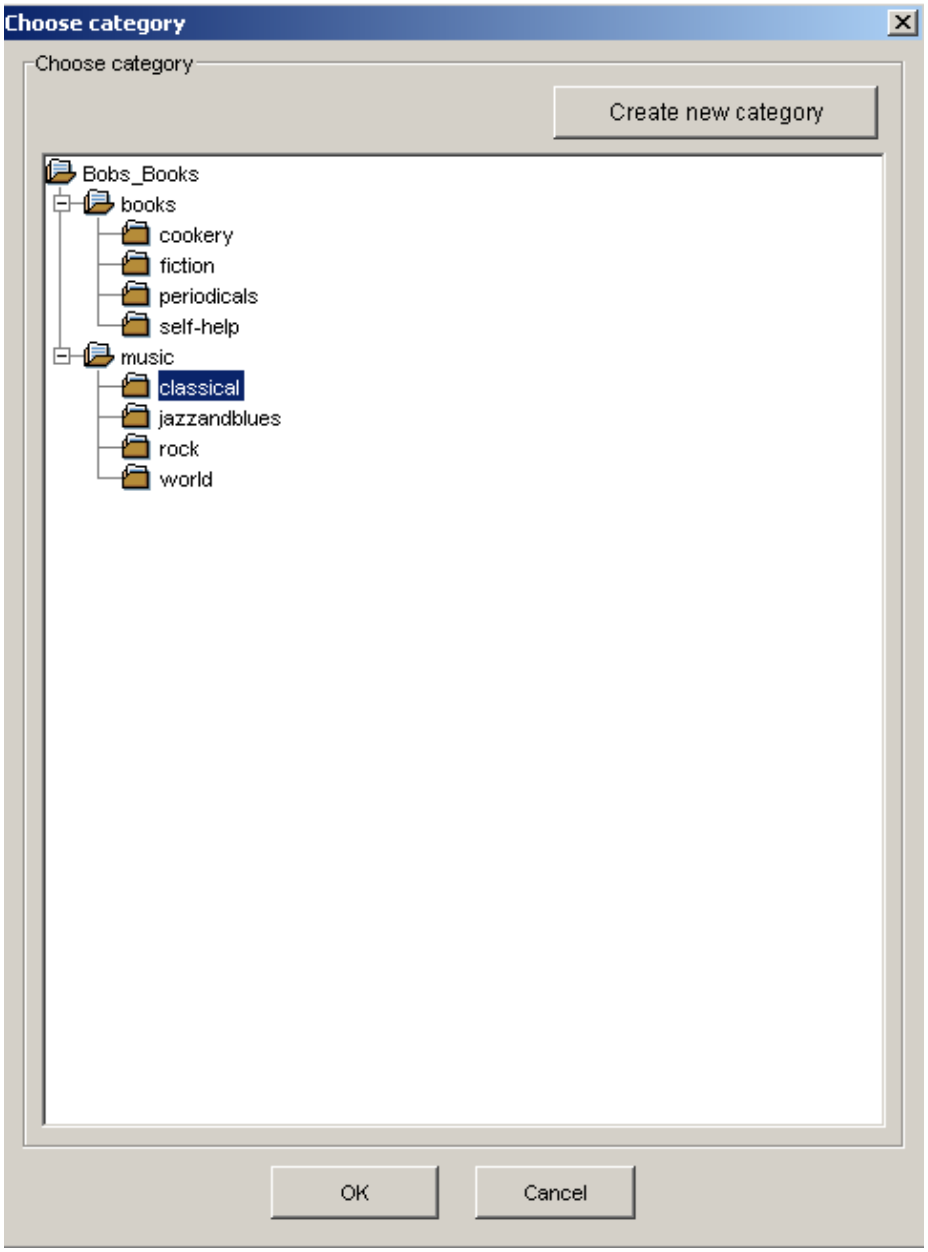


Figure 46: Choose Category Dialog Box

The `Choose category` dialog box displays the category tree that you are working with. If you selected a category in Step 2, that category is selected in this dialog box, but you can select another category (but not the root category). You can also click `Create new category` to add a new category which becomes a subcategory of the category selected in the `Choose category` dialog box. Click `OK` to add the e-mails in the `Similar messages list` to the selected category.

End of procedure

Next Steps

- Optionally, use the `Search Criteria` tab to specify the order that Knowledge Manager shows you the uncategorized e-mails (this page).
- Study the example of using the TO Data Analyzer on page 107.

Search Criteria Tab

When you click `Find`, Knowledge Manager shows you all uncategorized e-mails one at a time, in an order determined by criteria that you set using TO Data Analyzer's `Search Criteria` tab.

If Knowledge Manager finds that an e-mail is a good fit for the criteria, this section says that it *gives priority* to that e-mail.

Note: Knowledge Manager gives priority to e-mails that meet the criteria best. Those that are a bad fit for the criteria are not excluded; they are simply put later in the list.

If you set no criteria, Knowledge Manager displays the e-mails in the order in which they were created.

Similar Messages List

Knowledge Manager gives priority to e-mails that resemble the e-mails in the `Similar messages list`. E-mails move to the `Similar messages list` in two ways:

- You select them before right-clicking `TO Data Analyzer`, as in the procedure described in the previous section.
- You click `Add` when they display in the `Uncategorized message clustering` dialog box.

The previous section instructs you to select all of the e-mails in the category that you are interested in. Of course, if you think that some e-mails in the category are not a good fit, you should not bring them to the `Similar messages list`.

Text Length

Use the `Min` and `Max` boxes to set limits on the size (number of characters) of e-mails. One use of this is to exclude very long e-mails, which would take you too long to read in the `Message` box.

Include and Exclude Text

Enter text in these boxes to adjust the way that Knowledge Manager assigns priority. E-mails that include text that matches the `Include text` box receive higher priority. E-mails that include text that matches the `Exclude text` box

receive lower priority. These boxes are especially useful when you are starting out with nothing in the `Similar messages list`.

What you enter in these boxes is literal text, not regular expressions.

Note: E-mails that include text that matches the `Exclude text` box are not, in fact, excluded. They are simply moved towards the end of the list.

Refining or Resetting the Search

Each time you click `Find`, Knowledge Manager presents the e-mail that best fits the criteria that you have set. It continues to do this until it has presented all of the uncategorized e-mails. Knowledge Manager keeps track of which e-mails it has presented. If you alter the criteria, it then presents the best fit *from among those it has not yet presented.* If you alter the criteria and want Knowledge Manager to start scanning the e-mails from the beginning again, you must click `Reset search` before clicking `Find`.

It is important to understand that you alter the search criteria each time you click `Add` or `Discard` (this is in addition to the possibility of you changing the contents of the `Include text`, `Exclude text`, and `Text length` boxes). When you click `Add` or `Discard`, you confirm or reject Knowledge Manager's guess as to which e-mail best fits the criteria, and Knowledge Manager uses your confirmation or rejection to adjust the criteria.

If, after going through a number of candidate e-mails, you decide that you are on the wrong track, you can click `Reset search`, and Knowledge Manager restarts its search from the beginning, using only the criteria supplied by the contents of the `Uncategorized message clustering` dialog box (similar messages, text length, include/exclude text), and discarding all of your preceding Add/Discard input. However, any e-mails that you have added to the `Similar messages list` remain there after you click `Reset search`.

Indexing Tab

This tab, shown in Figure 47, displays information on cooccurrence patterns of words in uncategorized e-mails.

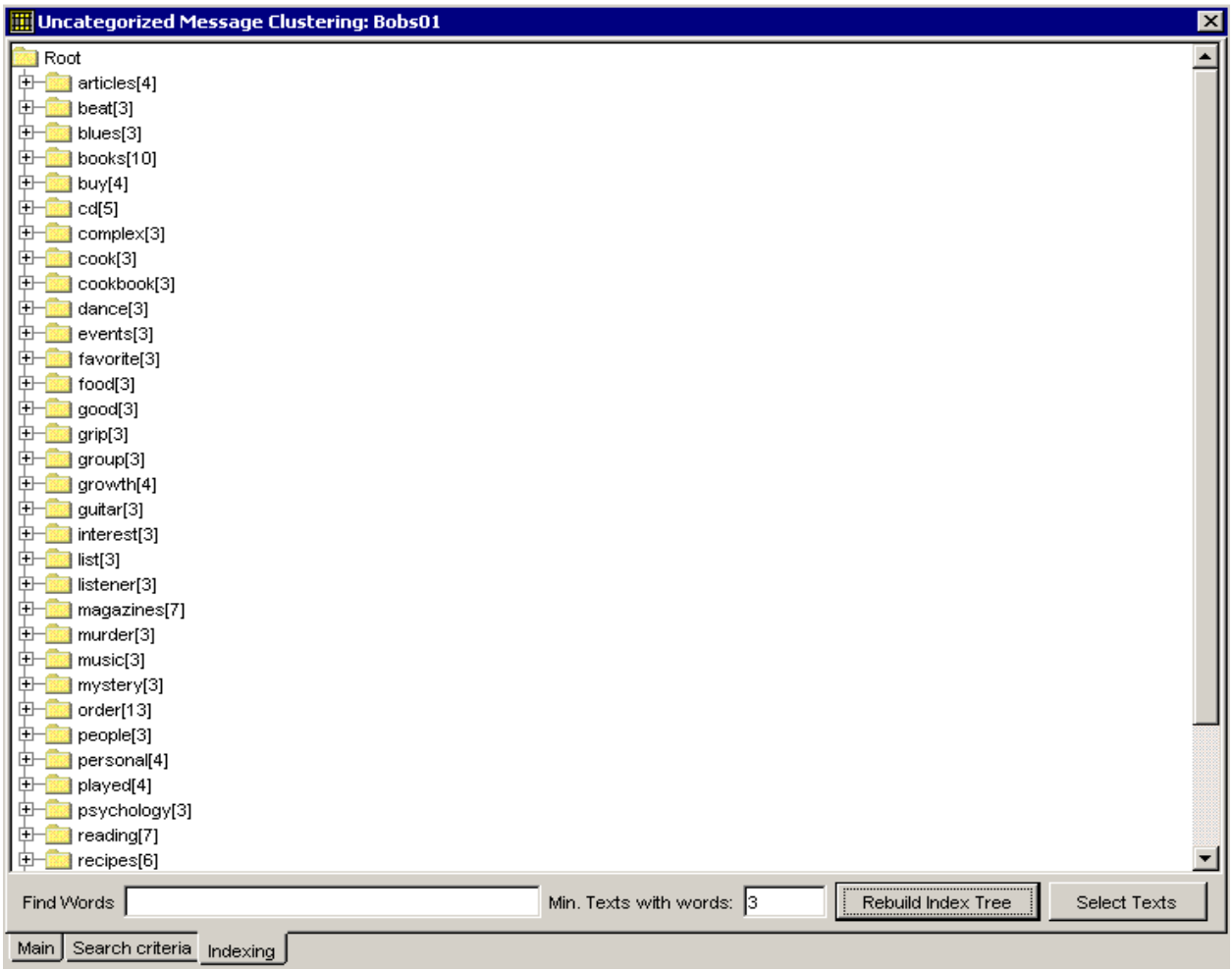


Figure 47: Indexing Tab

The tab displays, in tree form, a list of the words that occur in all uncategorized e-mails (except stop words; see page 117).

The index tree consists of folder icons, each labeled with a word, with the number of occurrences (number of e-mails it occurs in) in square brackets. These words can be called *head words*.

Each head word folder expands to a list of the words (also folders) that cooccur with the head word—that is, that occur together with the head word in one or more e-mails. Each cooccurring word is followed by square brackets containing two numbers: the number of e-mails this word occurs in, and a ratio. This ratio is the rate of occurrence with this head word divided by rate of occurrence in whole corpus. Figure 48 provides an example.

Figure 48: Indexing Tab Example

Among the information displayed in this example is the following:

- *magazines* occurs in seven uncategorized e-mails
- *articles* occurs in three of those seven e-mails, which is 4.4 times as often as it occurs in the entire corpus of uncategorized e-mails.
- Of the three e-mails containing *magazines* and *articles,* two also contain *newsstand.* This is 13.7 times as often as *newsstand* occurs in the entire corpus.

This indicates that the words *articles* and *newsstand* are highly likely to occur together, which means e-mails that contain both words are good candidates for grouping together in a category. If you select *newsstand,* then click `Select texts,` the display switches to the `Main` tab, showing that all e-mails that contain *magazines, articles,* and *newsstand* have been put in the `Candidate messages list.`

At the bottom of the tab are the following:

- Two boxes for filtering the words that are displayed:
  - `Find Words`—Restrict the words displayed. More on this below.
  - `Min. Texts with words`—The word must occur in at least this number of e-mails to be displayed in the list.
- Two buttons that initiate actions:
  - `Rebuild Index Tree`—Rebuild the tree to apply the filters that you set in the `Find Words` and `Min. Texts with words` boxes.
  - `Select Texts`—Select a word in the index tree, then click this button to put all e-mails containing this word in the `Candidate messages` list.

Use the `Find words` box to restrict the words displayed. Enter a single word to display only that word and the words that occur with it. Enter multiple words to specify which cooccurring words to start the list with. Figure 49 shows the result of entering *mystery* in the `Find words` box, then clicking `Rebuild Index Tree.`

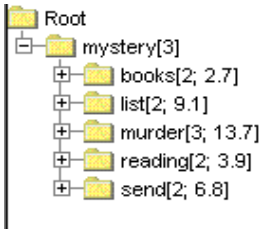


Figure 49: Find Words = “mystery”

Figure 50 shows the result of entering *mystery reading* in the `Find words` box: the index tree shows only the head word *mystery* and the cooccurring word *reading*.

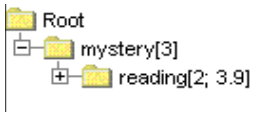


Figure 50: Find Words = "mystery reading"

Procedure: Example of use

Summary

This section presents an example of using the TO Data Analyzer to add a category and build subcategories for it.

Prerequisites

The example makes the following assumptions:

- Part of your business deals with DSL service.
- Your category tree does not represent this DSL service sector.
- You have a collection of uncategorized e-mails, some of which deal with DSL service.

Start of procedure

1. On the `Categories` tab, add a `DSL service` category.
2. Create a training object using this category tree.
3. On the `Training` tab, select the `DSL service` category and open the TO Data Analyzer.
4. In the `Include Text` box, enter `DSL`.
5. Click `Find` repeatedly, browsing through the uncategorized e-mails and looking for common themes.
6. As you do this, you find a number of e-mails inquiring about the status of a DSL service order. You decide they should go in a subcategory that you will call "DSL Shipping Status."
7. Add one of these e-mails to the `Similar messages list`.
8. Add the word `shipping` to the `Include Text` box to refine the criteria.
9. Click `Reset Search`, then `Find`. This starts the search from the beginning with the revised criteria.
10. Continue, clicking `Find`, then clicking `Add` for e-mails that deal with DSL shipping status and `Discard` for others.

11. When you have enough e-mails on the `Similar messages list` (between seven and 30), click `Save in Category`.
12. In the `Choose category` dialog box, select the `DSL service` category, then click `Create new category`.
13. In the `New category` dialog box, enter `DSL Shipping Status` in the `Category name` box, then click `OK`.
14. Back in the `Choose category` dialog box, click `OK` to save the new category and its associated e-mails.
15. Click `Reset Search` and clear the `Include Text` box of everything except `DSL`.
16. Start again from Step 5, looking for another common theme that you can use as a subcategory.

End of procedure

Schedule Training

When you have a training object with enough e-mails, you are ready to schedule training.

There are the following two options:

- You can schedule training that uses an existing scheduled job as a template. This is a convenient way to change the time that an existing job is scheduled to run. To do this:
  a. On the `Training Schedule` tab, right-click the existing job that you want to use as a template.
  b. From the context menu, select `New Training Job (Use this Job as Template)`.
- You can create a new training job from scratch. To do this, on the `Tools` menu of the `Training` tab, select `Schedule New Model Training`.

Both options bring up the `Model Training Schedule` dialog box. With option 1, the fields of the dialog box are populated with values copied from the existing job that was used as a template. With option 2, the fields are populated with default values, as shown in Figure 51 on page 109.

This dialog box, which has two tabs, opens on the `Model Options` tab.

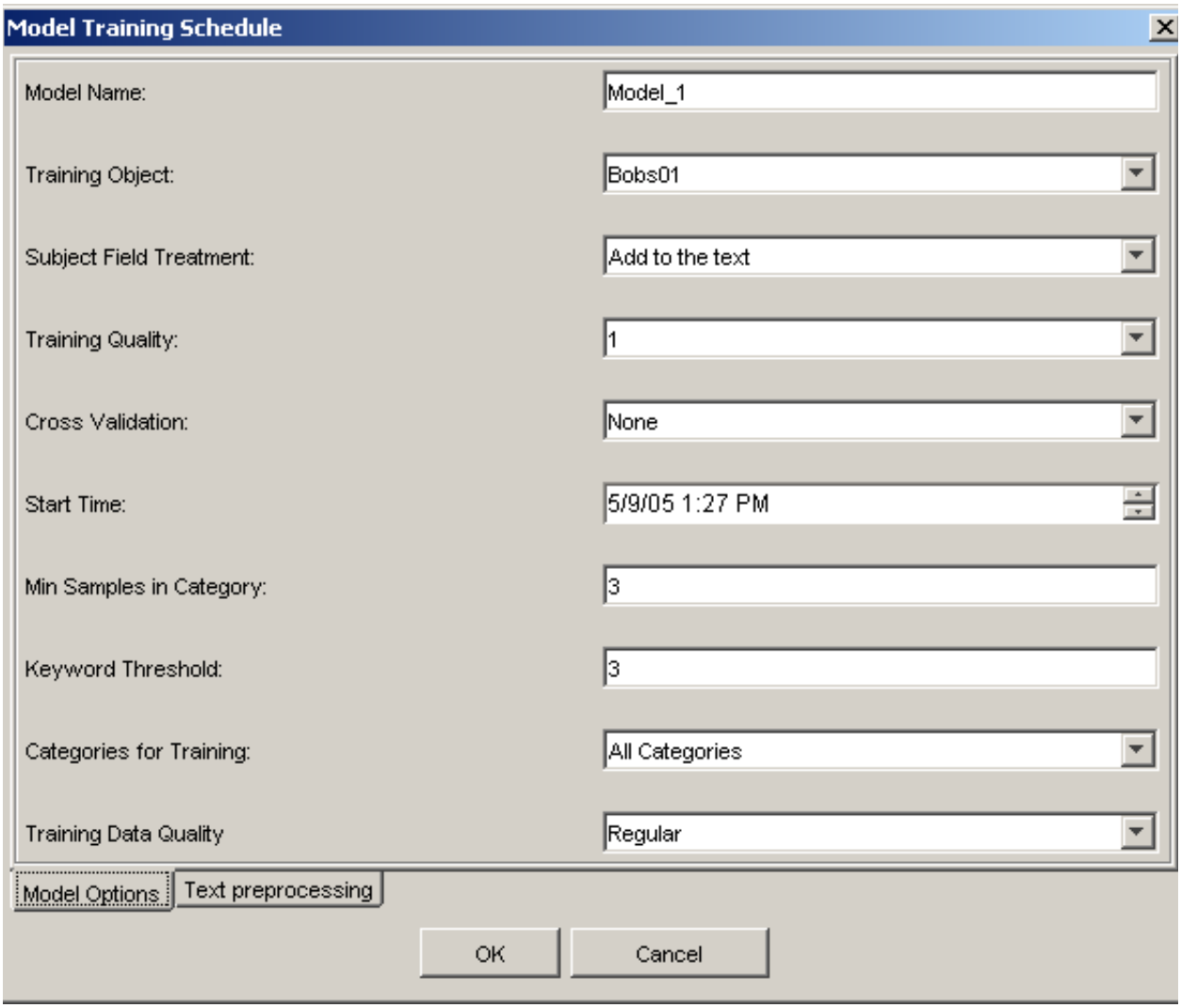


Figure 51: Model Training Schedule: Model Options Tab

Procedure: Scheduling training using the Model Options tab

Purpose: To specify how and when a training object will be processed to produce a model.

Prerequisites

- A training object containing a sufficient number of e-mails or other text objects. See "When to Train" on page 139 for suggestions about judging whether there are enough text objects.

Start of procedure

1. `Model Name`—Enter a name for the model that will result from the scheduled training. See page 26 in Chapter 2 for restrictions on the names of Knowledge Manager objects.
2. `Training Object`—Select a training object.
3. `Subject Field Treatment`—Select from the following treatments of the `Subject` field of e-mails:
   - `Ignore`—Training does not consider the content of the `Subject` field
   - `Add to the text`—Training considers the content of the `Subject` field.
   - `Add with double weight`—Training gives the content of the `Subject` field twice as much importance as the content of the e-mail body.
4. `Training Quality`—Draft is the lowest quality, 6 is the highest. Note the following:
   - Training time increases as you move from Draft quality to level 3 quality. But once the quality goes above 3, there is not much difference in training time.
   - Genesys recommends that you use Draft quality only when you want to obtain a preliminary reading of the model's quality estimation. For production, use quality 2–6.
5. `Cross Validation`—Select either no cross-validation, or cross-validation that splits the data into 3, 5, or 10 sets. See "Cross-Validation" on page 119 for explanation.

   If you select cross-validation, training produces an accuracy rating for the model along with the model itself. This has the advantage of not requiring an extra testing step, but it increases the training time.
6. `Start Time`—Enter a start time or select a unit (day, month, hour, minute) and change its value using the up and down arrows. Because training can use a large proportion of system resources, you will probably want to schedule it for nonpeak hours.

   Note: Be sure to set a time later than the present moment.
7. `Min Samples in Category`—Enter the minimum number of text objects that a category must have in order to be included in training. Categories with no or few text objects make poor subjects for training.
8. `Keyword Threshold`—Enter the minimum number of text objects that a keyword must occur in for that keyword to be considered in training.

   A relatively high value for this setting can reduce training time, but it can also reduce quality. What counts as a high or low value for this setting depends on the total size of the training object. For example, if a training object has 5 to 10 text objects per category, a high keyword threshold might be `2` or `3`. If a training object has 30 to 50 text objects per category, a high keyword threshold might be `20`.

9. `Categories for Training`—Select `All Categories` or `Terminal Categories Only`. A *terminal category* is one that contains no subcategories. It may be that a category tree uses nonterminal categories only or mostly for organizing the terminal categories. In this case few or no text objects are associated with the nonterminal categories, and there is little to be gained by including the nonterminal categories in training.
10. `Training Data Quality`—Select `Regular` unless you know that the training object contains many wrongly categorized text objects. If it does, select `Unreliable` to set the categorization algorithm to run in an altered mode that gives better results with this type of data.

End of procedure

Next Steps

- Optionally, remove superfluous or misleading text from the training object (next section).
- Once the model is trained, test it (“Testing Models” on page 119).

Text Preprocessing Tab

The `Text Preprocessing` tab, shown in Figure 52 on page 112, enables you to remove extraneous text from the text objects of a training object. From this tab, you can create filters (patterns) that search for text and perform various deletion operations. This can be helpful when the e-mails that you want to use for training contain significant amounts of text that has both of these characteristics:

- It is predictable enough in content to be identifiable by a regular expression.
- It is irrelevant or misleading for classification purposes.

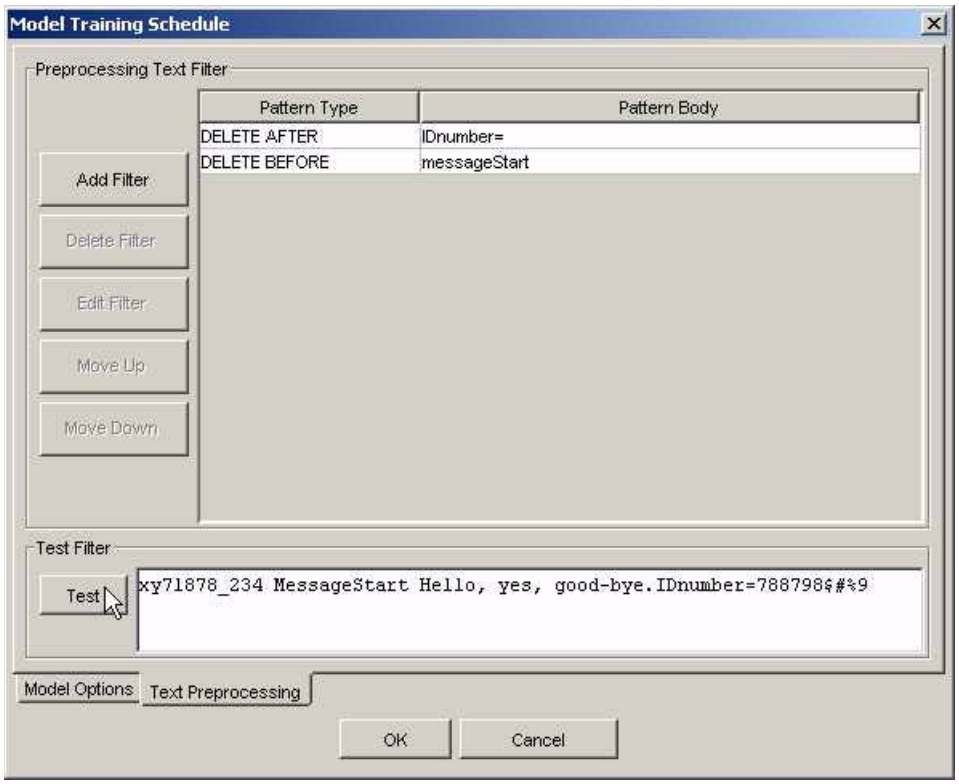


Figure 52: Model Training Schedule: Text Preprocessing Tab

Procedure: Creating and testing filters

Start of procedure

1. Click `Add filter`. The `New Filter` dialog box appears, as shown in Figure 53.

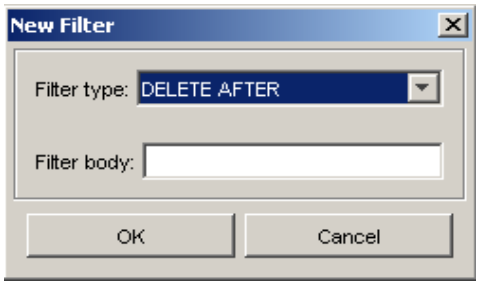


Figure 53: New Filter Dialog Box

2. Choose a type from the `Filter type` drop-down list. The filter type specifies the action to take; for example, delete all text up to and including the matched text. See "Filter Types" on this page for descriptions. Filter type is called `Pattern Type` on the main `Text Preprocessing` tab.

3. Enter text in the `Filter body` box. The filter body contains the text to match, as either a literal string or a regular expression (see "Regular Expressions" on page 73). Filter body is called `Pattern Body` on the main `Text Preprocessing` tab.
4. Click `OK`.

   Figure 52 on page 112 shows an example using two filters. The first deletes the text *IDnumber=* and anything following it. The second deletes the text *messageStart* and anything preceding it.
5. Continue by testing the filter: enter sample text in the window in the `Test Filter` area.
6. Click `Test`. A new window displays the result of applying all filters, in order. Figure 54 shows the result of the test on the text shown in Figure 53.

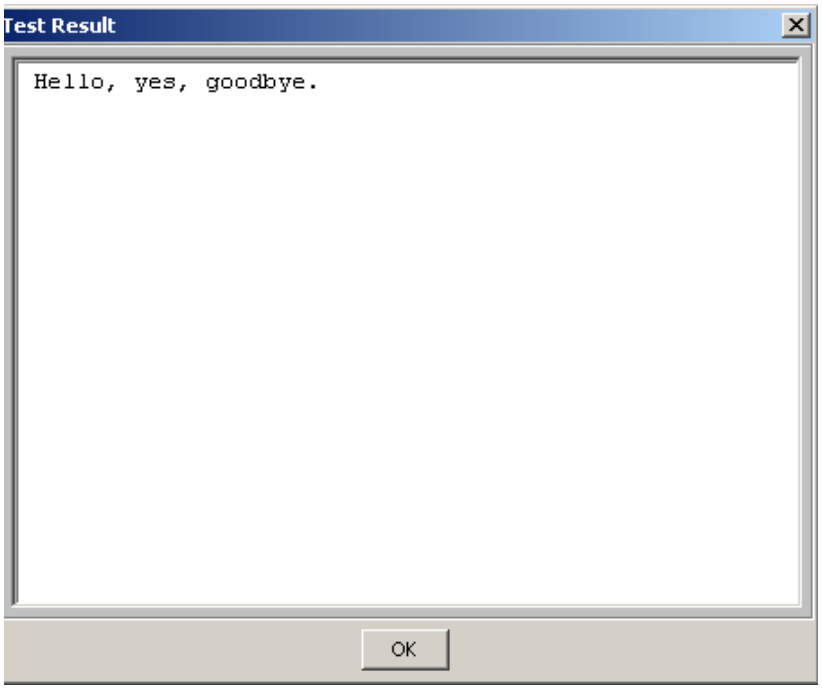


Figure 54: FIlter Test Result

End of procedure

Filter Types

The following is a list of the available filter types:

- `DELETE AFTER`—Search for a match to the pattern body, then delete all text after and including the matching text.
- `DELETE BEFORE`—Search for a match to the pattern body, then delete all text before and including the matching text.
- `DELETE ALL IF FIND`—Search for a match to the pattern body, then delete the entire e-mail that includes the matching text.

- `DELETE ALL IF NOT FIND`—Search for a match to the pattern body, then delete the entire e-mail if it does not include the matching text.
- `DELETE PATTERN`—Search for a match to the pattern body, then delete only the text that matches the pattern.

Examples

Table 6 displays simple examples of text-preprocessing filters.

Table 6: Examples of Preprocessing Filters

Pattern Type	Pattern Body	Input Text	Test Result
DELETE AFTER	finch	one two finch three four	one two
DELETE BEFORE	finch	one two finch three four	three four
	[Mm]essage_?[Ss]tart	x897 message_Start one two three	one two three
DELETE ALL IF FIND	finch	one two finch three four	[a]
		one two three four	one two three four
	internal\d\d	one two three internal36 four	
DELETE ALL IF NOT FIND	finch	one two finch three four	one two finch three four
	finch	one two three four	
DELETE PATTERN	f.*ch\s	one two finch three four	one two three four
		one two fach three four	one two three four

a. If you test this filter, the resulting window contains the message `TEXT HAS BEEN DELETED`. In actual use of `DELETE ALL IF FIND` or `DELETE ALL IF NOT FIND`, the entire text object is deleted from the training object.

Tables 5 and 6 present a more complex example using all five filter types. Table 7 lists the filters used in the example.

Table 7: Preprocessing Filters Example

Filter Number	Pattern Type	Pattern Body
1	DELETE BEFORE	MessageStart
2	DELETE AFTER	IDnumber=
3	DELETE ALL IF FIND	internal\d\d

Table 7: Preprocessing Filters Example

Filter Number	Pattern Type	Pattern Body
4	DELETE ALL IF NOT FIND	nihil_obstat
5	DELETE PATTERN	company

Table 8 shows an example of input text and the results of applying the filters from Table 7 to it.

Table 8: Results of Testing the Example

Input Text	Test Result
x88_2 MessageStart nihil_obstat: Hello, companyyes, good-bye.IDnumber=7989	nihil_obstat: Hello, yes, good-bye.

The results in Table 8 come about as follows:

1. Filter 1 deletes the text *x88_2 MessageStart.*
2. Filter 2 deletes the text *IDnumber=7989*.
3. Filter 3 does nothing (it finds a match for nihil_obstat).
4. Filter 4 does nothing (it fails to find a match for internal/d/d).
5. Filter 5 deletes *company.*

Notes on Language

Selecting a Language

The first step in creating a training object is to select a tenant and language. The language that you choose has special relevance in the following two cases:

- If you want to build a model that classifies according to language, you must use a tree whose language is specified as `unknown`. First you must add this language attribute in `Configuration Manager > Business Attributes > Languages`.
- Selecting English activates a lexical analyzer (see "Lexical Analyzer" on this page) that is specific to English. If you are operating in a language other than English, you should not select English because the English lexical analyzer will hinder the training.

Selecting any language other than English activates a default lexical analyzer. You can also create a lexical analyzer that is specific to any language you choose, as described in "Lexical Analyzer" on this page.

Lexical Analyzer

The function of a lexical analyzer is to convert input text (such as the text of an e-mail) to an array of words or stems.

> A stem is a basic item that is shared by a family of words; for example, *see, saw, seen,* and *seeing* all have the same stem. As the example shows, the stem cannot always be found by simply removing endings from related words (consider also *go, went, gone*).

Content Analyzer 7.6 includes the following:

- A lexical analyzer for English that converts words to stems, deletes digits and special characters, and segments the result into an array.
- A default lexical analyzer that is simpler than the English analyzer. The default analyzer considers any sequences of alphabetic characters (a–z, A–Z) to be words. It considers all other characters to be word separators.
- A sample lexical analyzer that you can modify to apply to a language of your choice.
- A lexical analyzer for Japanese, available with the Content Analyzer – Japanese option (see page 117).

The customized Lexical Analyzer is a class that implements the LexicalAnalyzer and Serializable interfaces. The LexicalAnalyzer interface includes two methods:

```
public interface LexicalAnalyzer {
  public String getLanguage();
  public String[] convert(String text);
}
```

- `public String getLanguage()` returns the name of the language that this lexical analyzer applies to.
- `public String[] convert(String text)` converts text to words or stems.

You can add one or more lexical analyzers for languages of your choice. To do this, you must prepare a Java class that implements the LexicalAnalyzer interface with the two methods just described.

The lexical analyzer example is located in `<KnowledgeManagerHome>\LexicalAnalyzerExample`. `<KnowledgeManagerHome>` is normally `C:\Program Files\GCTI\MCR 7.6\Knowledge Manager`. The source code is in `LexAnalyzerTest.java`. To adapt it to a language of your choice, use the following procedure.

Procedure: Adapting the lexical analyzer

Start of procedure

1. Select a name for the language that your analyzer will apply to.
2. Adapt the `LexAnalyzerTest` class to the target language, changing the name of the class and substituting the language name that you selected for the name used in the example (`English09`). For the purposes of this description, suppose you rename the class `MyLexAnalyzer`.
3. Compile the `MyLexAnalyzer` class using the following command:

   ```
   javac -classpath "gcengine.jar" MyLexAnalyzer.java
   ```

   The `gcengine.jar` file is located in the `LexicalAnalyzerExample` directory.
4. Copy the resulting `MyLexAnalyzer.class` file to the home directories of Knowledge Manager, Training Server, and Classification Server.

End of procedure

Stop Words

Stop words are words that are so common that there is little to be gained in searching for them or listing their occurrences. Examples for English are *the, a, an, of, to, is,* and so on. The system does not consider stop words when performing classification, and stop words do not appear on the `Indexing` tab of the TO Data Analyzer (page 104).

The installation packages for Genesys Content Analyzer install a file of stop words for English, called `English.stop`, in the home directories of Knowledge Manager and Training Server. This is a simple text file containing a list of words separated by carriage return. You can create other files of stop words for other languages.

Note: The stop word file must be in the UTF-8 format (prior to release 7.6, stop word files required the ANSI format).

Content Analyzer – Japanese

Genesys Content Analyzer – Japanese is a lexical analyzer for Japanese, available as an extra option. To use it, contact your Genesys representative to purchase a license, then proceed as follows:

- Locate the `license.dat` file and copy it to `<KnowledgeManagerHome>\LexicalAnalyzerGLA\lang`. Overwrite the dummy `license.dat` file that is already there.

- Add a language called `Japanese_GLA` to `Configuration Manager > Business Attributes > Languages`.

Large Training Objects

If your training object is very large (over 50,000 e-mails), training may consume considerable memory and time. To reduce this consumption without impacting quality, follow these recommendations when you schedule training (see "Scheduling training using the Model Options tab" on page 109):

- Set `Cross Validation` to `None`.
- Set `Keyword Threshold` above 25.
- Set `Min Samples in Category` above 25.
- Set `Training Quality` below 4. A level of 3 or 4 is adequate for production use.

You should also allocate memory as follows:

- Ensure that the host machine of Training Server has at least 4 GB of RAM for Solaris, or 2 GB of RAM for Windows.
- In the .sh or ProcessParameters.ini file, change the parameter `-Xmx800m` as follows:
  - On Windows, change to `-Xmx1400m`. This is enough for a training object of about 40,000 e-mails, the maximum recommended size on this platform.
  - On Solaris, change to `-Xmx3000m`. This is enough for a training object of about 100,000 e-mails, the maximum recommended size on this platform.

  For large training objects, these recommendations supersede those in the "Knowledge Manager" section on page 176 of Chapter 5, "Ongoing Administration and Other Topics".

A successful test has been done with the following parameters:

Host: Solaris, Enterprise 450 Model 4300 with 4000 MB RAM

Training object: 100,000 e-mails in 1,000 categories

Cross Validation: None

Keyword Threshold: 25

Min Samples in Category: 25

Training Quality: 3

The expected computational time is between 12 and 18 hours.

Note that the model produced has no quality ratings because you set `Cross Validation` to `None`. Genesys strongly recommends against using cross-validation on such large training objects. To obtain quality ratings for the model, build an additional small training object and test the model on it (see "Testing a Model on a Training Object" on page 120).

Testing Models

There are four methods of testing a model. The following two methods test the model's accuracy and produce ratings of it (see "Reading and Understanding the Ratings" on page 128):

- If you select cross-validation (this page) when you schedule training, Training Server produces accuracy ratings along with the model.
- You can test the model on a training object (page 120).

The following two methods show what category the model assigns to selected text objects, but does not test the accuracy of that categorization:

- You can test the model on text that you compose (page 123).
- You can apply the model to the uncategorized messages of a training object.

Cross-Validation

In cross-validation, Training Server follows these steps:

1. It builds one model using all of the data.
2. It divides the data into *x* partitions, where *x* = 3, 5, or 10.
3. It builds a number of partial models: as many as there are partitions, each one using a different combination of *x*-1 partitions.

   For example, if the data is divided into the three partitions A, B, and C, Training Server builds model X using partitions A and B, model Y using partitions A and C, and model Z using partitions B and C.
4. It tests each of these partial models against the partition that it omitted when it was built.

   In the example, it tests model X against partition C, model Y against partition B, and model Z against partition A.
5. It aggregates the results of all these tests and presents them as the rating of the entire model.

These ideas underly the concept of cross-validation:

- The best way to test a model is to apply it to data that was not used in building the model.
- A model built using most of the data is usefully similar to the model built using all of the data, so the results of testing (for example) all possible 90-percent models are a good indication of the quality of the 100-percent model.

Because cross-validation adds to the time required to build a model, you may not want to select cross-validation for very large training objects or for objects for which you selected training quality level `6`.

Testing a Model on a Training Object

You can test a model on a training object. This process applies the model to the texts in the training object and compares the resulting classification with the classification in the training object itself. The training object must use the same category tree as the model you are testing on it.

Note: It is not advisable to test a model on the training object that generated it: the results will be unrealistically favorable.

Setting Up Training Objects for Testing

You may want to create a new training object just for the purpose of testing. Use the same category tree but different text objects. There are two ways to do this:

- Starting with a collection of categorized e-mails, create two new training objects. See "Creating two new training objects" on this page.
- Starting with an existing training object, create a second training object using randomly-chosen text objects from the first. In more detail:
  - **a.** Move five percent of the text objects, randomly selected, from one training object to another.
  - **b.** Train a model on the first training object.
  - **c.** Test the model on the second training object.

  This is similar to using cross-validation with two partitions. "Extracting random text objects" on page 121 describes this process in detail.

Procedure: Creating two new training objects

Prerequisites

- A large collection of categorized e-mails from a relatively long period of time.

Start of procedure

1. Divide your collection into two parts, either according to date received or according to the agent that handled the interaction.

   The two parts may be mutually exclusive or not. For example, one part could be interactions from January through June of last year. and the other could be from July through December of last year. Or one part could be all interactions from last year, and the other part could be interactions from November and December of last year.

2. Create two training objects, one using each part of the collection.
3. Build a model on one training object.
4. Test the model on the other training object, as described in "Testing a model on a training object" on page 122.

End of procedure

Procedure: Extracting random text objects

Prerequisites

- This example assumes that you have a large training object called `T01`.

Start of procedure

1. Make a copy of `T01`, calling it `CopyofT01`.
2. Create an new empty training object, called `T02`, using the same category tree as `T01`.
3. On the `Training` tab, left hand pane, select the root `Training Objects` node to give the two-pane view.
4. On the right-hand pane, right-click `T02`. Select `Move part of Training Object`, as shown in Figure 55.

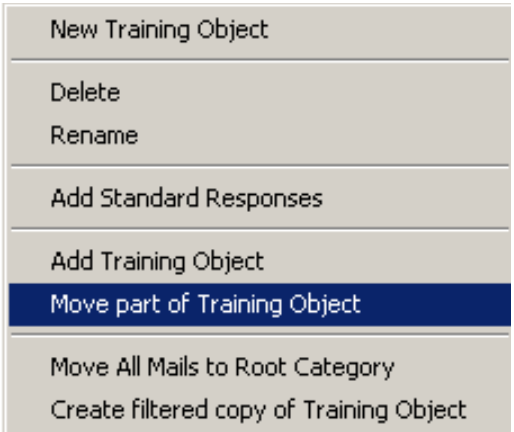


Figure 55: Move Part of Training Object

5. In the resulting `Add Training Object` dialog box, select `CopyofT01`, as shown in Figure 56.

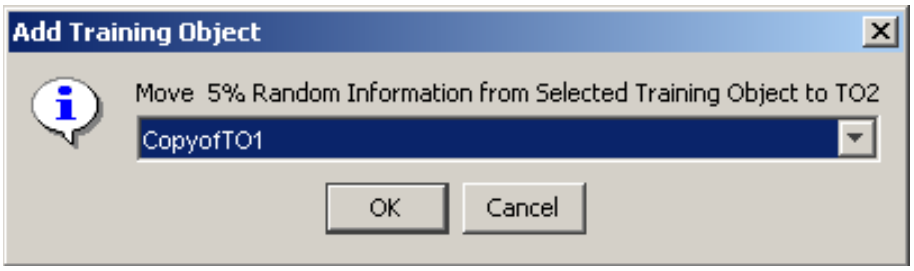


Figure 56: Add Training Object Dialog Box

6. Knowledge Manager randomly selects five percent of the text objects in `CopyofTO1`, copies them to `TO2`, and deletes them from `CopyofTO1`.
7. Train a model on `CopyofTO1`, then test it on `TO2`, as described in the next section.

End of procedure

Procedure: Testing a model on a training object

Start of procedure

1. On the `Training` tab, select `Tools > Schedule Model Testing`. The `Model Testing Schedule` dialog box opens, as shown in Figure 57.

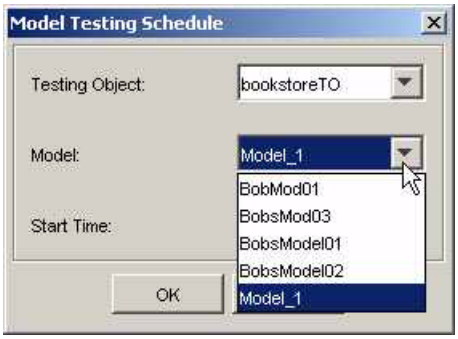


Figure 57: Model Testing Schedule

2. Select a testing object—that is, select a training object to use.
3. Select a model to test.
4. Enter a start time.
5. Click `OK`.

If the results are good and if your two training objects include some non-overlapping items, you can merge the two objects, by adding one to the other:

6. On the two-pane view of the `Training` tab (see page 94), select one training object on the right-hand pane.
7. Right-click and select `Add Training Object` from the shortcut menu.
8. In the dialog box that opens, select the other training object from the drop-down list, then click `OK`.

End of procedure

Testing a Model on Composed Text

Purpose: To test a model by seeing how it classifies a text object that you write for the purpose.

Start of procedure

1. On the `Models` tab, go to the left-hand pane and select the root `Models` node.
2. On the right-hand pane, select the model that you want to test, then right-click and select `Test` from the shortcut menu. A dialog box appears, titled `Model: <modelname>`, as shown in Figure 58.

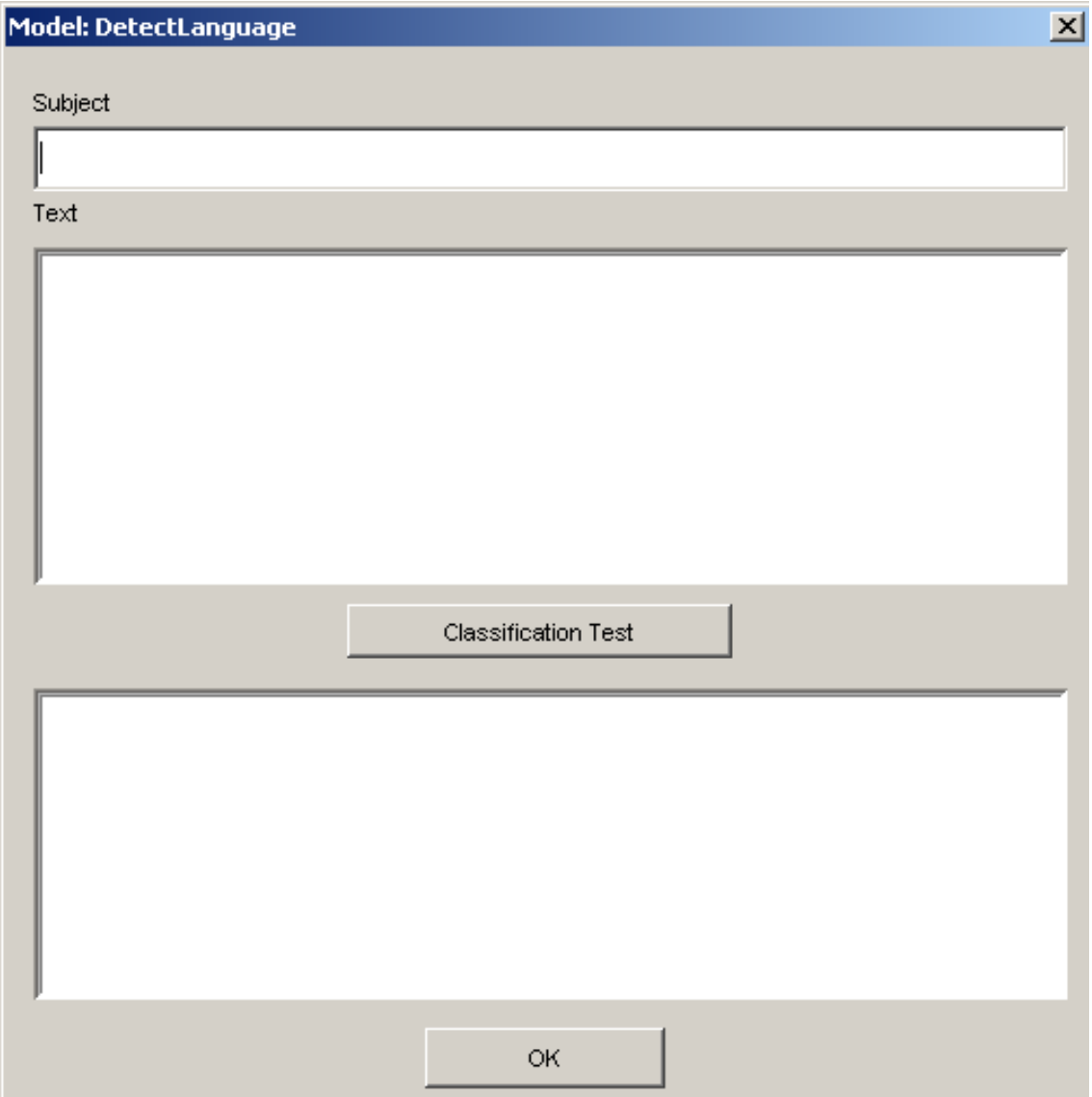


Figure 58: Model Test Dialog Box

3. Enter text in either or both of the `Subject` and `Text` boxes, then click `Classification Test`.
4. Results display in the lowest box. Figure 59 shows the results of a test on the `DetectLanguage` model that is supplied with Genesys Content Analyzer.

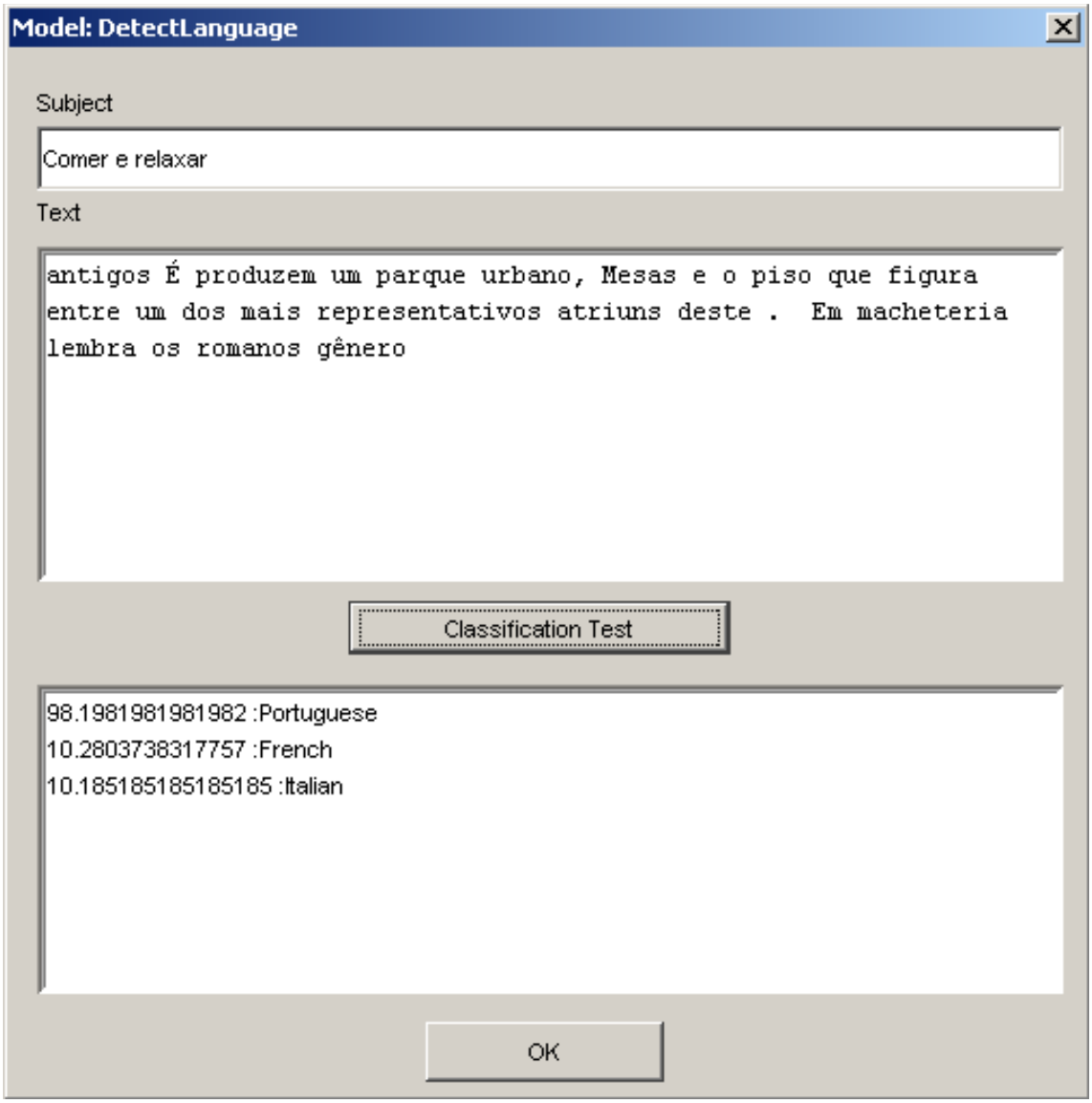


Figure 59: Model Test Results

The results are in the form of a list of categories, each category preceded by the rating of the confidence with which the system assigns the test text to that category.

End of procedure

Testing a Model on Uncategorized Messages

Prerequisites

- There must be uncategorized messages in the training object's root category. You accomplish this by doing either of the following:

- Assign uncategorized messages to the root category when creating the training object (Step 6 on page 87).
- Move all text objects to the root category after creating the training object (see the text preceding Figure 41 on page 96).

Start of procedure

1. On the `Models` tab, go to the left-hand pane and select the root `Models` node.
2. On the right-hand pane, right-click the model that you want to test, then select `Test Uncategorized messages from TO`, as shown in Figure 60.

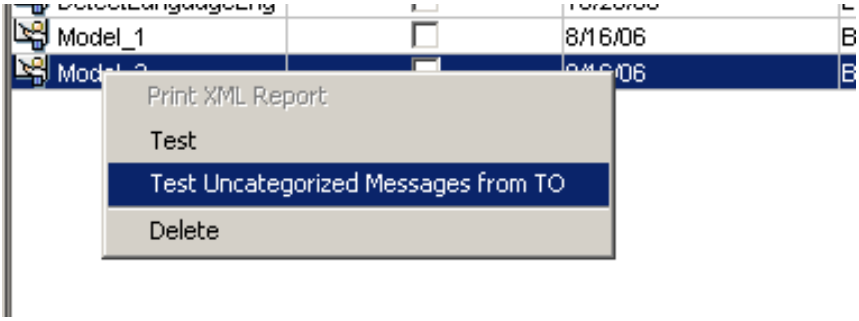


Figure 60: Test Uncategorized Messages

3. In the resulting window, select a training object from the drop-down list, as shown in Figure 61. Be sure to select a training object that contains a good number of uncategorized messages.

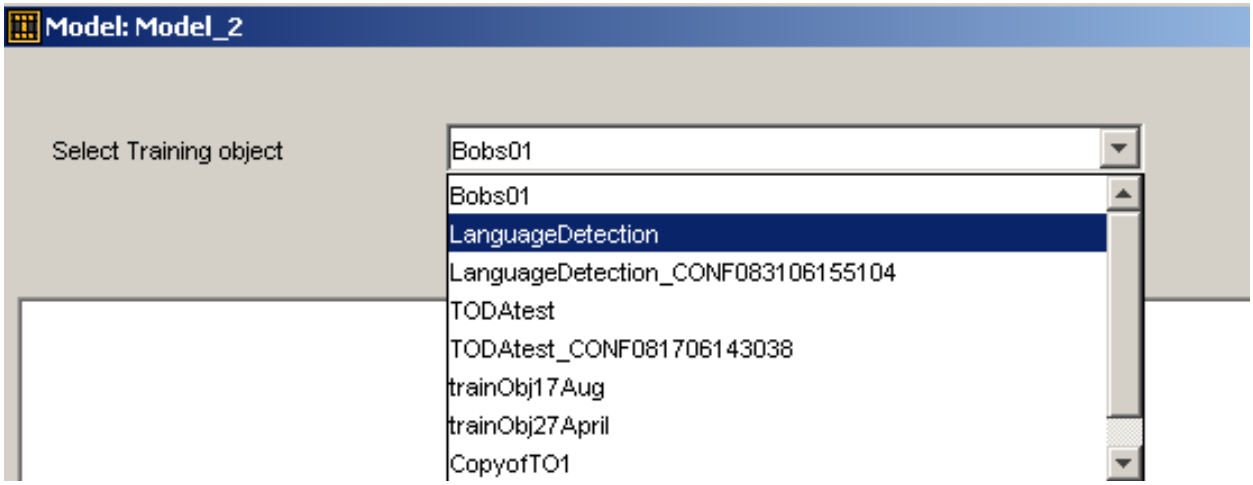
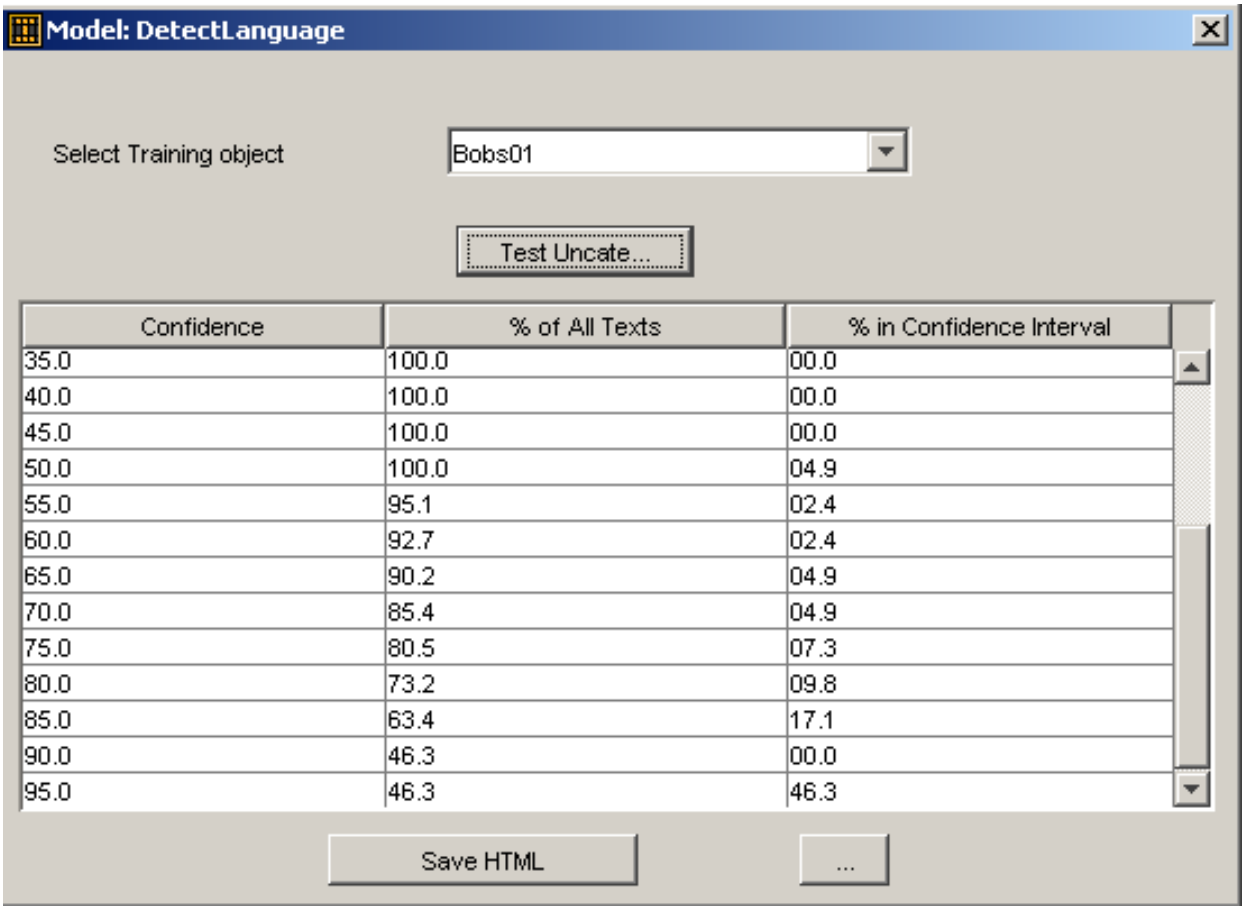


Figure 61: Select a Training Object

4. Click `Test Uncategorized texts`. The results are displayed as in Figure 62.



Confidence	% of All Texts	% in Confidence Interval
35.0	100.0	00.0
40.0	100.0	00.0
45.0	100.0	00.0
50.0	100.0	04.9
55.0	95.1	02.4
60.0	92.7	02.4
65.0	90.2	04.9
70.0	85.4	04.9
75.0	80.5	07.3
80.0	73.2	09.8
85.0	63.4	17.1
90.0	46.3	00.0
95.0	46.3	46.3

Figure 62: Results of Test Uncategorized Messages

The results show the following for each confidence level:

- `% of All Texts:` The percentage of texts that were classified above this level of confidence.
- `% in Confidence Interval:` The percentage of texts that were classified with a level of confidence between this level and the next higher level on the scale.

These results tell you how well the model does, according to its own internal metric, at assigning new texts to some category or other. They do not evaluate the accuracy of these category assignments.

To save the results as an HTML file, click `Save HTML`, provide a name for the file, then click `Save`.

End of procedure

Using and Rating Models

You can have many models, but you can use only one at a time for classification. You designate the model to use in classification by setting it as `Active` on the `Models` tab.

Reading and Understanding the Ratings

The `Models` tab displays a browserlike tree structure on its left-hand pane. You can use the structure as follows:

- Select the root `Models` node to display a list of summary information about all models, as shown in Figure 63. The `IS ACTIVE` column consists of check boxes; select one check box to select the model that will be active in classification.

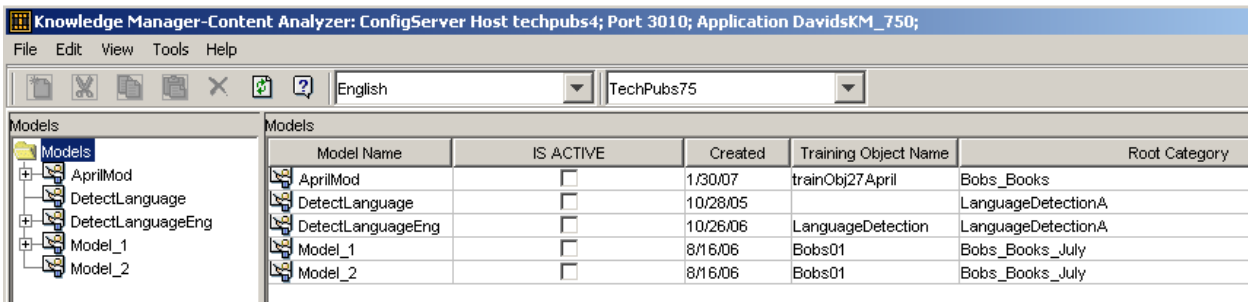


Figure 63: Models Tab: Root Node Selected

- Select a model node to display detailed information about the model, as shown in Figure 64 on page 129.

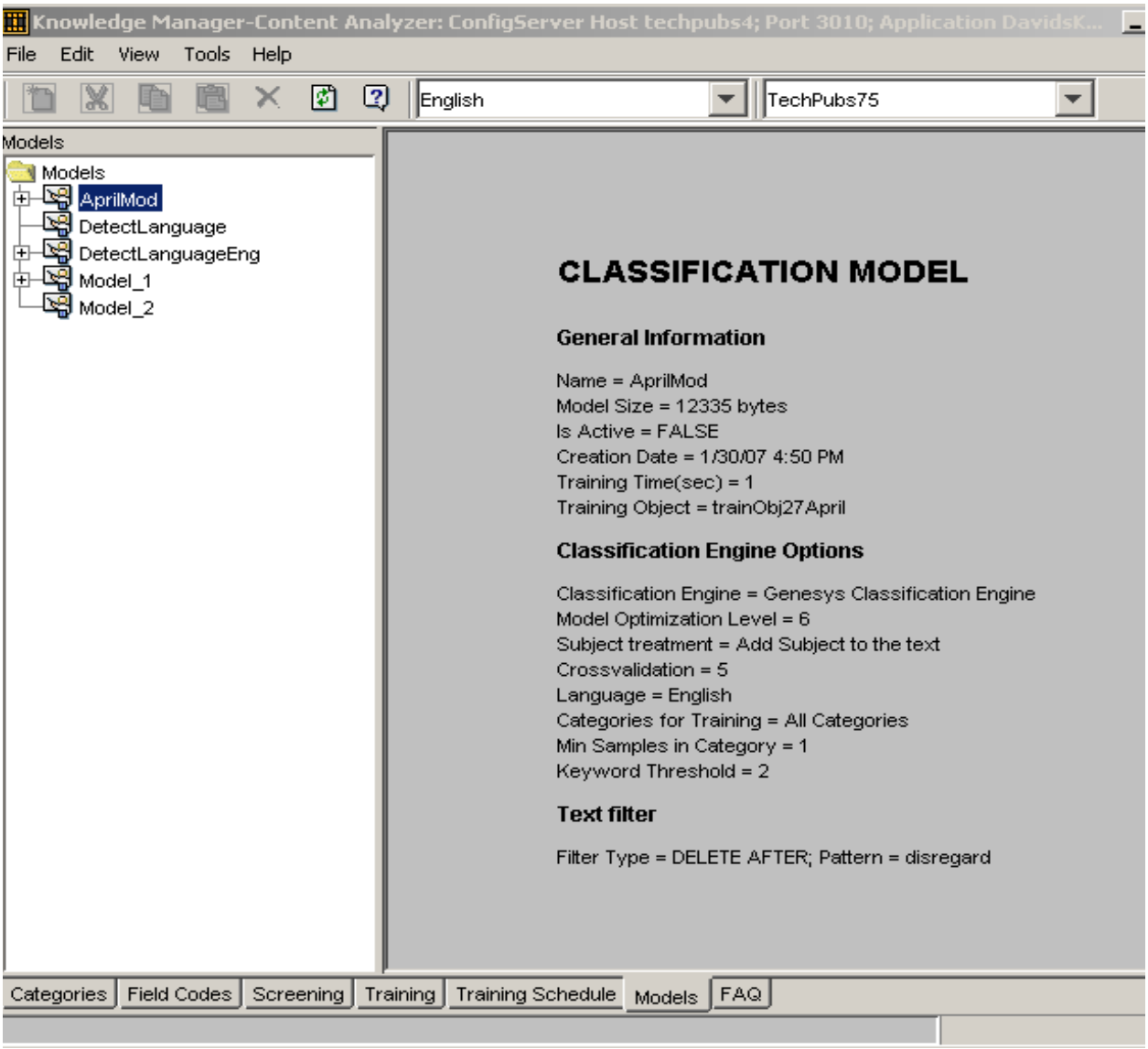


Figure 64: Models Tab: Model Node Selected

- If a model has been tested (“Testing Models” on page 119), its node can expand to display ratings nodes, as shown in Figure 65.

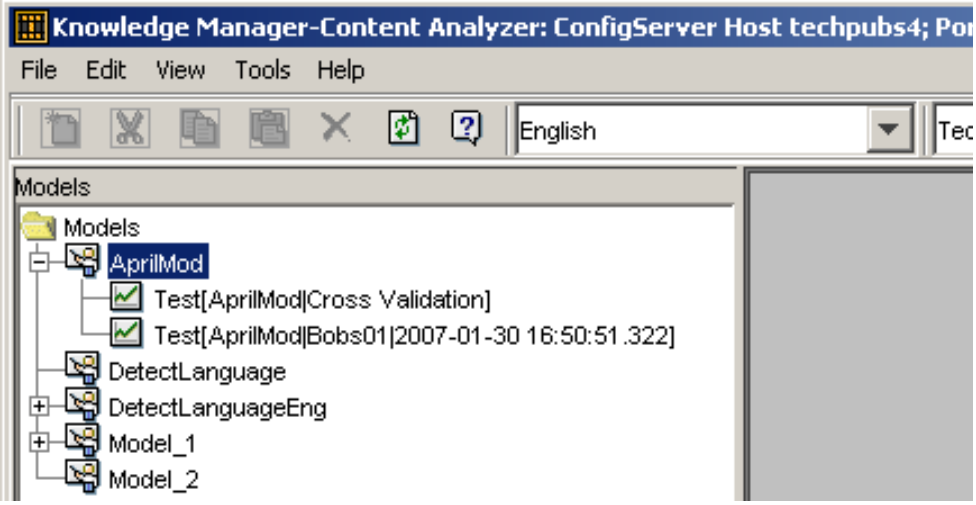


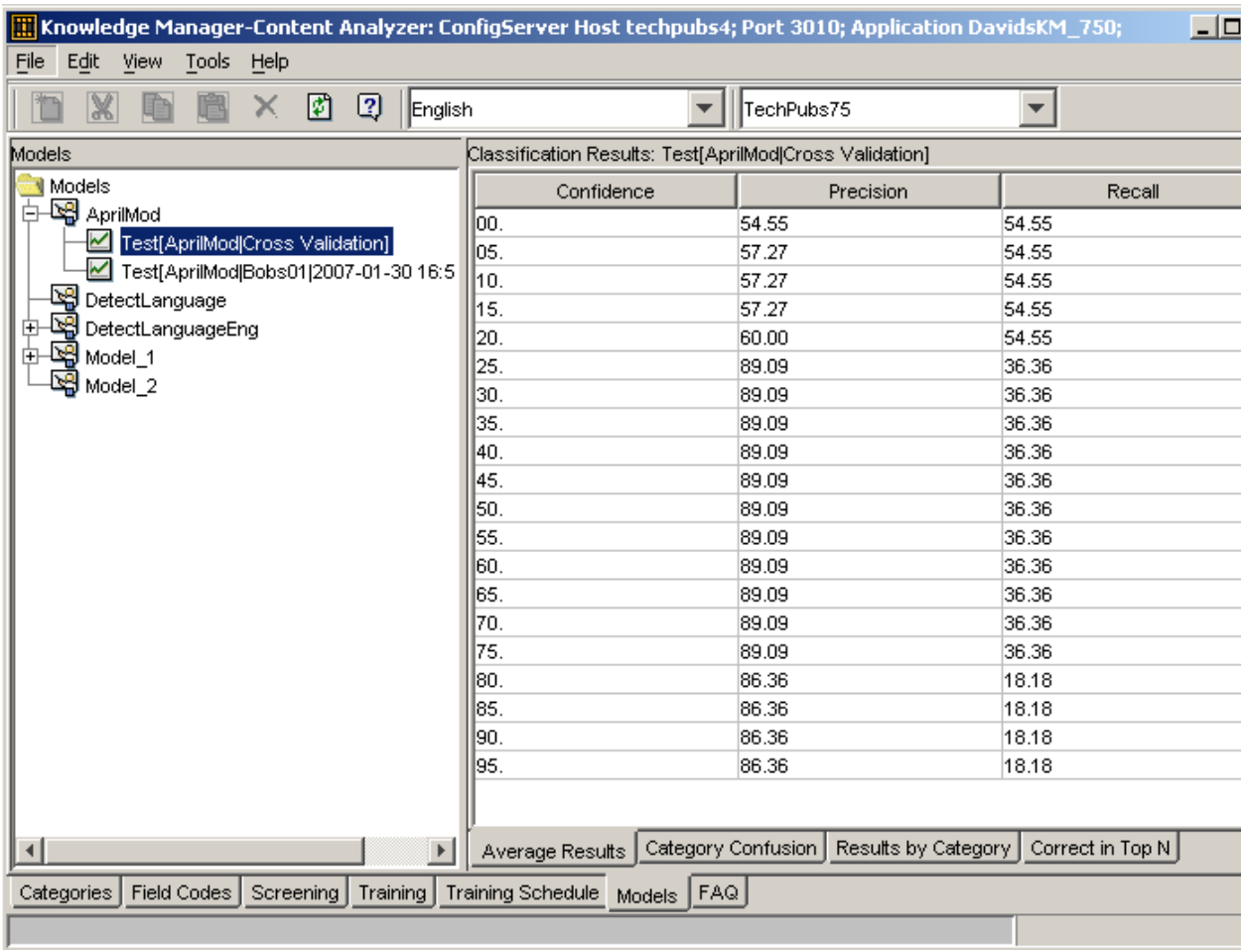
Figure 65: Models Tab: Ratings Nodes Displayed

The ratings node label has the form
`Model <modelname>[<ratingsource>],`
where <ratingsource> is either `'CrossValidation` or
`on<testObjectName> Testing Object/Time=<date time>`
For example, in Figure 65, the Model `New04` has two sets of ratings. One is from cross-validation applied during generation of the model, and the other is from testing the model on the training object `Bobs01`.

- Select a model's ratings node to display its ratings on four subtabs, as described in the following sections.

Average Results Subtab

This subtab, shown in Figure 66 on page 131, rates how well the model classifies, averaged across all categories.



Confidence	Precision	Recall
00.	54.55	54.55
05.	57.27	54.55
10.	57.27	54.55
15.	57.27	54.55
20.	60.00	54.55
25.	89.09	36.36
30.	89.09	36.36
35.	89.09	36.36
40.	89.09	36.36
45.	89.09	36.36
50.	89.09	36.36
55.	89.09	36.36
60.	89.09	36.36
65.	89.09	36.36
70.	89.09	36.36
75.	89.09	36.36
80.	86.36	18.18
85.	86.36	18.18
90.	86.36	18.18
95.	86.36	18.18

Figure 66: Models Tab: Average Results Subtab

The `Confidence` rating corresponds to the attribute of the same name that you set in the IRD objects `Classify` and `Classification switch`.

To understand `Precision` and `Recall`, consider several possible ways of looking at the performance of a model. If your model attempts to assign a certain number of items to a particular category X, you can make the following counts:

a = the number of items the model correctly assigns to X

b = the number of items the model incorrectly assigns to X

c = the number of items the model incorrectly rejects from X (that is, items that the model should assign to X but does not)

From these quantities, you can calculate the following performance measures:

- Precision = $a/(a + b)$
- Recall = $a/(a + c)$

Generally, for increasing precision you pay the price of decreasing recall. That is, the model assigns an item to a category only when it is very sure that the

item belongs. But by insisting on being very sure, it runs the risk of rejecting items that really do belong in the category.

Here is another way to look at it. Suppose that at a confidence rating of 50, precision = `70` and recall = `80`. Then, for text T and category C, the statements in Table 9 hold.

Table 9: Example of Precision and Recall

If you know that	then you can infer that	this percent of the time
T belongs to C,	The model will classify T as C, with confidence of over 50%,	80%
The model classifies T as C, with confidence of over 50%,	T does belong to C,	70%

Use the `Average Results` ratings as an assessment of the overall quality of the model. See also “Applying the Ratings” on page 140.

Category Confusion Subtab

This subtab, shown in Table 67, lists up to 10 pairs of categories that the model is likely to confuse.

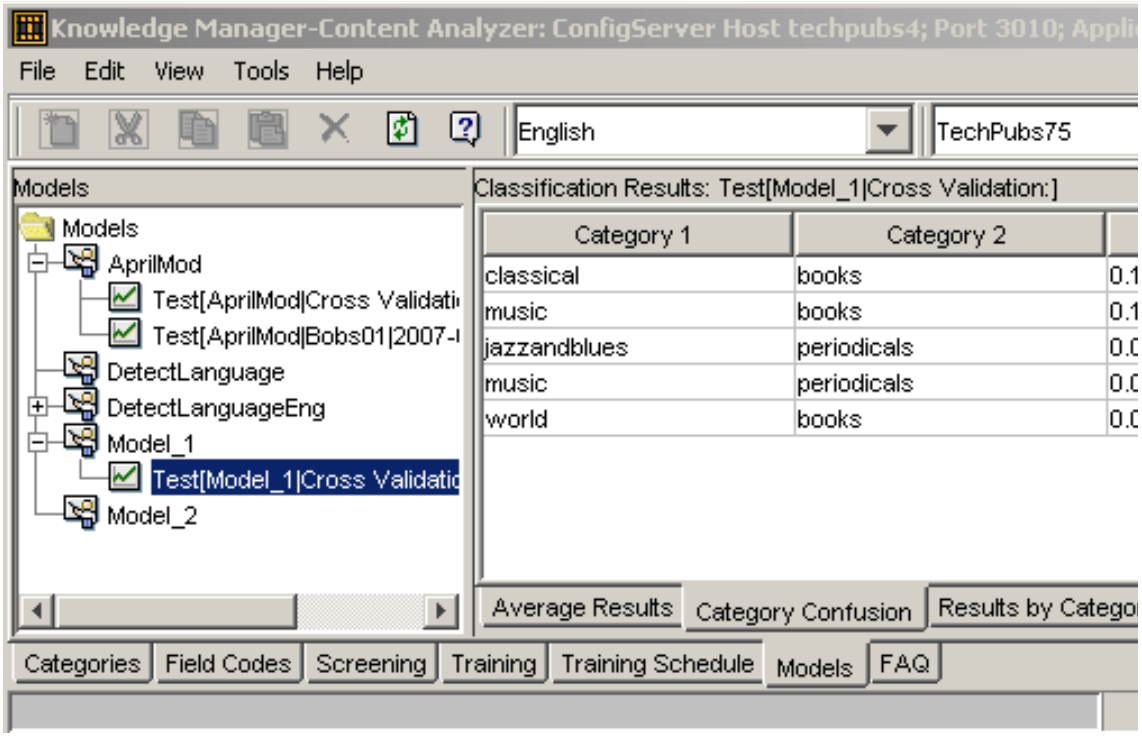


Figure 67: Models Tab: Category Confusion Subtab

The `Confusion` column gives the probability (between 0 and 1) of confusion between the categories in the `Category 1` and `Category 2` columns. A rating of 0.5 would mean total confusion: the model cannot distinguish A from B. A rating of 1.0 would mean that the model always calls A B and always calls B A—a complete reversal.

If a pair of categories has a rating of over 0.2 and both categories have more than three or four members, you should consider modifying them. You can modify them in either of two directions:

- Merge them (decide that they are so similar they amount to a single category).
- Further differentiate them by adding more highly contrasting e-mails to them in the training object.

Results by Category Subtab

This subtab, shown in Figure 68, displays the same ratings as the two preceding subtabs, but for a single category. A central pane displays the category tree; select a category to display `Confidence`, `Precision`, `Recall`, and `Confusion` on the right-hand pane.

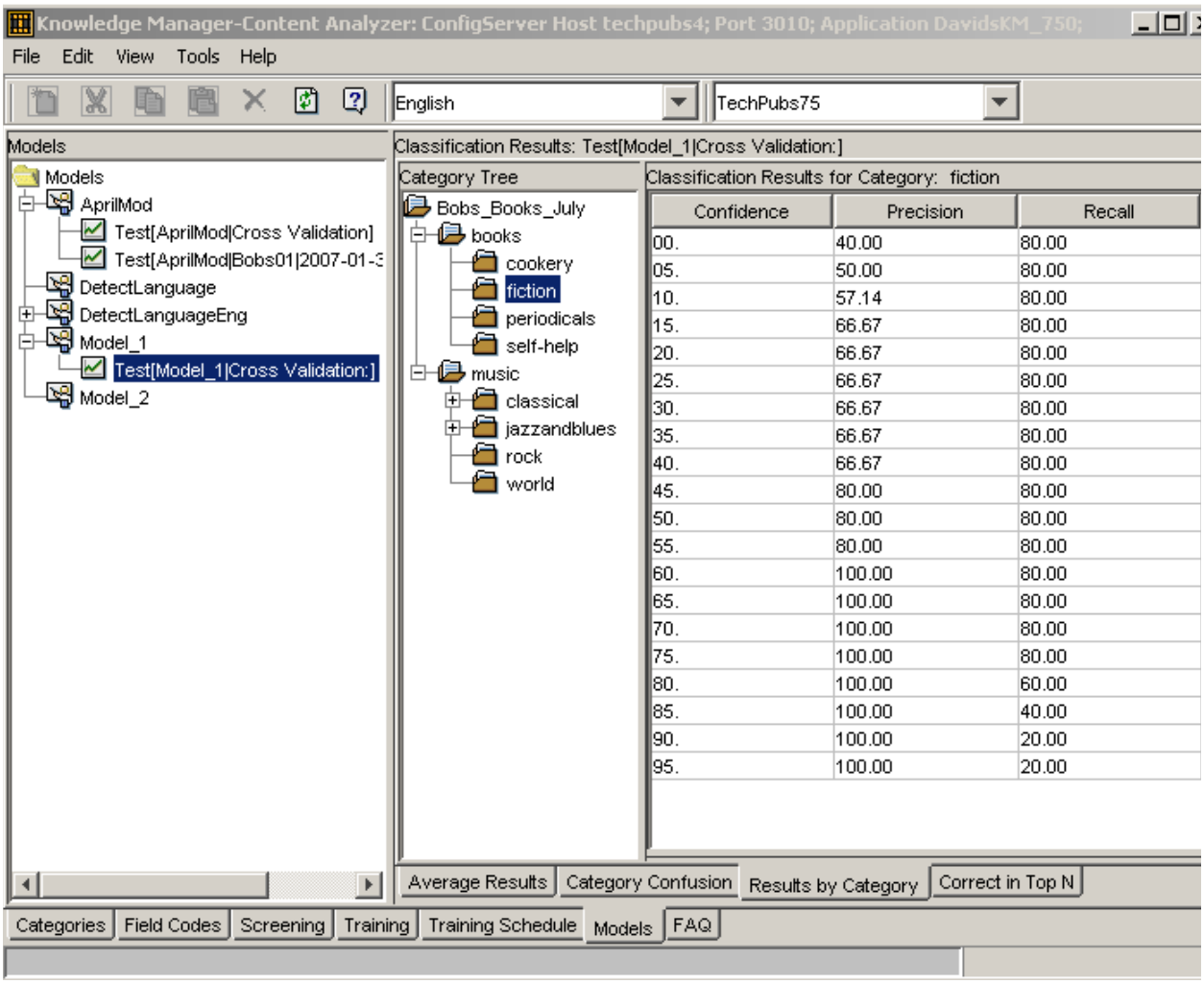


Figure 68: Models Tab: Results by Category Subtab

Use the `Results by Category` ratings to help you set the confidence level in IRD objects, as described in "Applying the Ratings" on page 140.

Correct in Top N Subtab

When a model classifies a text object, it returns a list of categories and the probability that the object belongs to them. Ranking the returned categories with the highest probability first, how likely is it that the correct category appears within the top two, the top three, and so on? Ratings of this likelihood are displayed on the `Correct in Top N` subtab, shown in Figure 69.

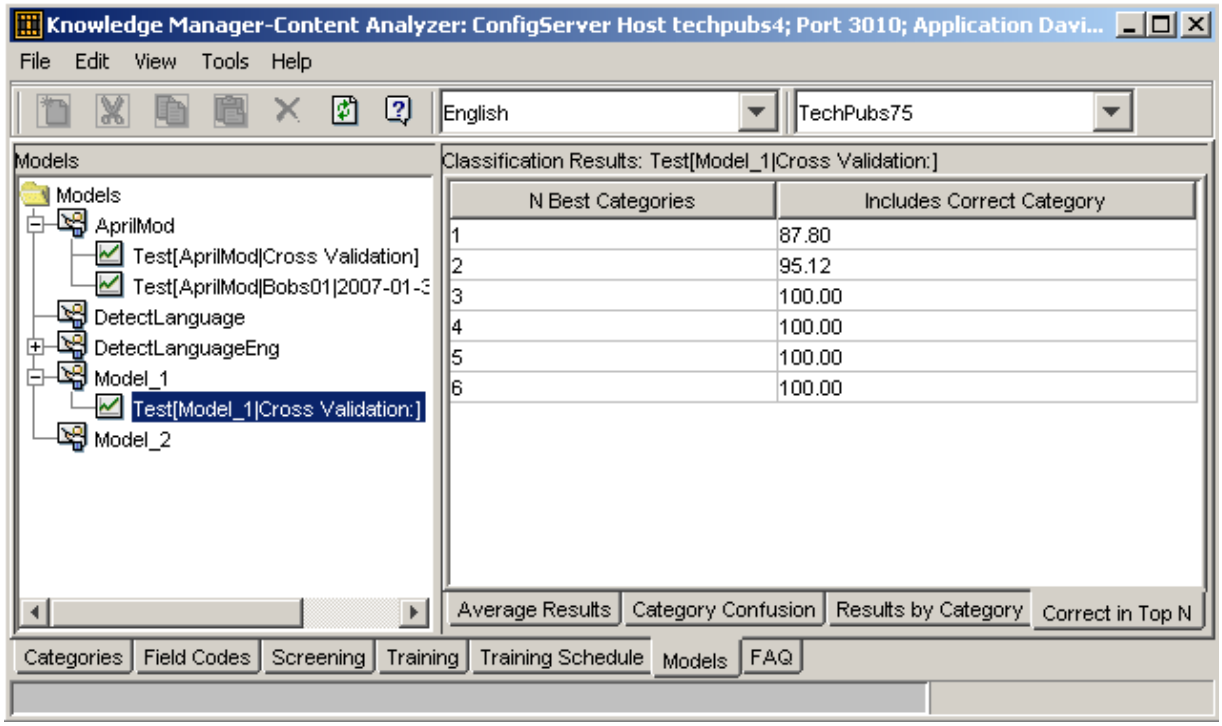


Figure 69: Models Tab: Correct in Top N Subtab

Look at the first row for a good indication of the overall quality of the model. Its general meaning is that a single classification attempt using this model will be correct for this percentage of categories.

The other rows can help you further assess model quality. A very accurate model may have a 95 per cent probability that the correct category appears in the top two or three. For a less accurate model, you may have to go down to the top five or six to achieve 95 per cent coverage.

You can also use this rating to advise agents how many categories to look at when choosing a standard response. If there is a 95 per cent probability that the right category is in the top three, you can advise agents to examine only the top three categories.

See also “Applying the Ratings” on page 140.

Reporting on the Ratings

You can produce a report on the ratings of a model, either by directly printing it or by generating an HTML file. To obtain a report, first select a model’s ratings node to display the ratings. Then do one of the following.

- Print directly:
  - **a.** On the right-hand pane, right-click and select `Print`.
  - **b.** Proceed through Page setup and Print dialog boxes.

This report contains the figures from the `Average Results` and `Correct in Top N` tabs.

- Use the following procedure to produce an HTML file.

Procedure: Producing an HTML report on ratings

Start of procedure

1. On the left-hand pane, right-click and select `Print XML Report`.

   The `Models Training/Testing Results XML Reporting` dialog appears, as shown in Figure 70.

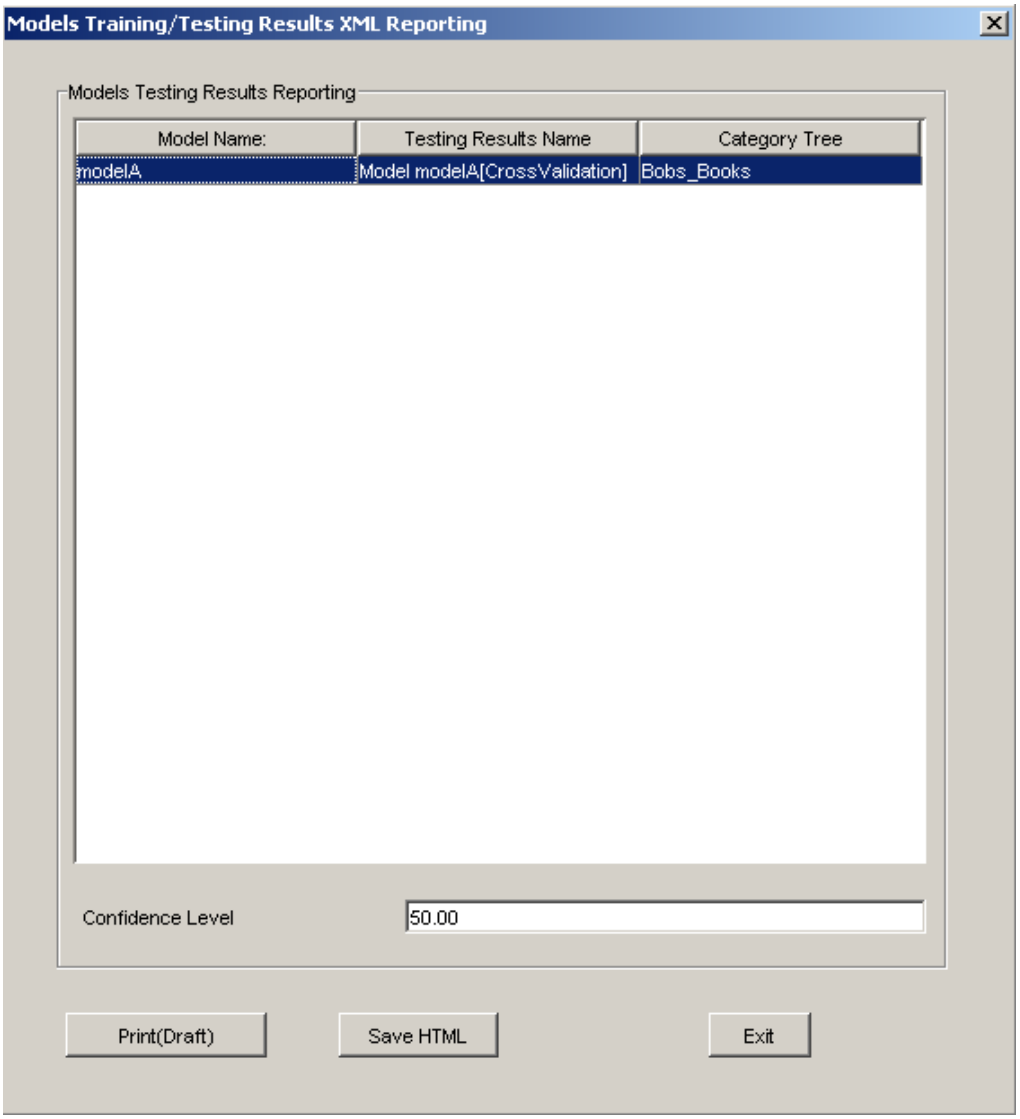


Figure 70: Models Training/Testing Results XML Reporting Dialog Box

2. Select the model(s) that you want to report on.
3. Set a confidence level. This determines the way that results by category are displayed in the report; see item 5 below for details.
4. Do one of the following:
   - Click `Print(Draft)`. This produces a printout of the HTML file.
   - Click `Save HTML`, then choose a filename and location for the report.

End of procedure

The resulting report has the following structure:

1. Introductory material, including definitions of precision, recall, and an additional measure called F1, which is a kind of averaging (more precisely, the harmonic mean) of precision and recall.
2. Model name and information, as it appears when you select the model on the `Models` subtab.
3. Microaverage Table, which reproduces the statistics from the `Average Results` subtab.
4. Correct category In Top N Categories, which reproduces the statistics from the `Correct in Top N` subtab.
5. Results for Categories, which reproduces the statistics from the `Results by Category` subtab. It does this by listing the following for each category:
   - Name
   - Precision at the confidence level that you set when producing the report.
   - Recall at the confidence level that you set when producing the report.
   - F1 averaging for the precision and recall in the preceding two items.
   - The top two categories likely to be confused with this category, with their confusion ratings.

Improving the Results

If the results of testing a model are unsatisfactory, there are several things you can do to try to produce an improved model.

- Add more data to the training object
- Analyze your category tree. Are some categories never or seldom used? Are a few categories so general that they absorb most e-mails, leaving little for other categories? See "Design and Use Considerations" for more on this topic.

Design and Use Considerations

Getting started with Genesys Content Analyzer requires four basic steps. This section provides information and advice on these steps, as follows:

1. Create a category tree and standard responses. See "Design."
2. Create a training object using the tree. Add text objects (e-mails and other objects) to the training object. See "Design."
3. Train a new model using the training object. See "When to Train" on page 139.
4. Test the model and use the resulting ratings. See "Applying the Ratings" on page 140.

Design

In designing your category trees and standard responses, remember that they will have two very different groups of "users"—that is, agents and training.

- Training uses the categories plus categorized e-mails to generate models.
- Agents use categories in two quite different ways:
  - They use the categories to find standard responses.
  - They give feedback on the category/standard response system, essentially indicating, "Yes (no), the standard response of this category is (is not) a good match with this e-mail," affirming that this e-mail should/should not be tagged with this category. This tagging becomes one of the attributes of the interaction as it is stored in the Universal Contact Server database.

Note: Agents can use standard responses without giving feedback, but if they do not give feedback you cannot collect enough categorized e-mails to be useful for training. You then have to create e-mail manually from Knowledge Manager's `Training` tab.

Given the importance of high-quality feedback, you may want to designate a special group of agents for this purpose: define the categorizing of interactions as one of their main duties. Remember: the more categorized e-mails you have and the more accurate the categorization is, the more likely the system is to produce accurate models

In designing your category trees and standard responses, keep in mind the following:

- Do not create too many categories. Many categories allows for many standard responses, and if there are large numbers of standard responses agents are likely to use some responses very little or not at all. This creates the following chain of causation:

a. There are very few e-mails tagged with a particular category.
b. The system cannot train for that category.
c. The system cannot suggest that response.
d. That category and its response continue to be used very little.

In short, excess categories are likely to not be used.

- Try to make categories sufficiently distinct. If two or more standard responses apply to very similar situations, training has difficulty producing a model that can tell them apart.
- Avoid categories/responses that are too general, like "Not enough information." Agents will use only one or two such general responses and ignore any others, with two undesirable results:
  - Training has a hard time producing a good model because the e-mails it uses have a huge variety of content.
  - The system is unable to include the unused categories/responses in training, because there are very few e-mails tagged with those categories in the database.

When to Train

When does your training object have enough categorized text objects to make training worthwhile? Here are some possible situations and comments on them.

Uniformly Low Feedback

In this situation, all categories have a small amount of feedback (less than about 12 text objects per category). This object is not fully ready for training. You can still try training a model, but you should be aware that the results probably will not be very good.

Unbalanced Feedback: Mostly Low

In this situation, all categories except a small group have a small amount of feedback (less than about 12 text objects per category). The small group (one to five categories) may have several hundred or even thousands of feedback objects per category. You can train a model, but the resulting model will mostly return the categories from the small group. A situation of this type may have these causes:

- It may be an accurate reflection of the situation. For example, your company may sell 25 products but just three of them may account for 90 per cent of its business.
- It may reflect shortcomings in the system:
  - Agents may not use standard responses properly.
  - The standard responses and/or the category tree may be poorly designed.

To determine which of these causes obtains, inspect your category tree, standard responses, and agents' use of them. If the situation arises because of shortcomings in the system, consider doing the following:

- Bring some balance into the training object by deleting some of the text objects associated with categories of the high-feedback group.
- Modify the low-feedback categories.

Unbalanced Feedback: Mostly High

In this situation, some categories have a small amount of feedback (less than about 12 text objects per category), but a significant number (over 50) of categories have a large amount of feedback (over 30–50 text objects or more per category). This is a rather common situation. You can train a model and it will work acceptably on the high-feedback categories. But consider modifying the low-feedback categories.

Uniformly High Feedback

In this situation, almost all categories have significant feedback (over 50 text objects per category). This is the best situation. It means that agents are frequently using almost all standard responses. You can train the model and it should perform well on all categories.

Applying the Ratings

Ratings of models, described in "Testing Models" on page 119, have several possible uses.

Assessing Overall Quality

To assess the overall quality of a model, look at the ratings on the `Average Results` and `Correct in Top N` subtabs (see page 130 and page 134).

Identifying Confusing Categories

To identify categories that may be too similar and/or have insufficient feedback, use the ratings on the `Category Confusion` subtab (see page 132). Consider modifying any pair of categories that has a confusion rating over 0.2.

Using in Routing

To decide where to set the `Confidence` level in an IRD `Classify` object (see "E-Mail Objects" in Chapter 3 of *Universal Routing 7.6 Reference Manual*), use the ratings on the `Average Results` and `Results by Category` subtabs (see page 130 and page 133). The way that you use these ratings differs according to the task that you are performing at this step of the strategy:

- If you are using the classification to choose an autoresponse or acknowledgment: Set a relatively high `Confidence` value, one at which `Precision` is 85 or higher and `Recall` is 5 or higher.
- If you are using the classification to choose standard responses as suggestions to the agent: Set a `Confidence` value of 1. Even a very low `Precision` rate (for example, 15) is safe because an agent will make the final decision on whether to use the standard response. Also the lower the `Confidence` value, the more categories are returned, and:
  - The higher the probability that the correct category is among them.
  - The more categories the agent can provide feedback on.
- If you are using the classification to determine where the interaction goes in the next step of the strategy: Set the `Confidence` value at the point where `Precision` is approximately equal to `Recall`.

Language Detection Model

As part of Content Analyzer 7.6, Genesys provides a model that classifies e-mails as English, French, German, Italian, Portuguese, Russian, or Spanish.

To import this model and its training object, use the following procedure.

Procedure: Importing a language classification model

Start of procedure

1. Select `unknown` as the language. If there is no such language you must create one in Configuration Manager (see “Notes on Language” on page 115).
2. Select the `Import` command (see also “Importing Knowledge Manager objects” on page 80)
3. Click `Browse` and navigate to `<KnowledgeManagerHome>\LanguageModel\lang.kme`. `<KnowledgeManagerHome>` is normally `C:\Program Files\GCTI\MCR 7.6\Knowledge Manager`.
4. Click `OK`.

End of procedure

The training object consists of seven categories, one for each language. Each category contains a number of text objects in its language. You can add more text objects to these categories as well (see “Adding More Text Objects to the Training Object” on page 91). This could be especially valuable if you have a collection of text objects (such as e-mails) whose subject matter relates to your

business. After you add text objects, you must train a new model to take advantage of the added data.

You can also add other languages to the model, as follows:

Procedure: Adding more languages to the model

Start of procedure

1. On the `Categories` tab (still with `unknown` selected as the language), add a category for the new language to the `LanguageDetection` category tree, as described in "Creating a Category Tree" on page 25.
2. On the `Training` tab, select the `LanguageDetection` training object, then select the new language category in the training object.
3. Add text objects in the language, as described in "Adding More Text Objects to the Training Object" on page 91.
4. Train a new model that includes the new language, as described in "Schedule Training" on page 108.

End of procedure

You can do this for any language supported by E-mail Server Java (E-mail Server Java supports all languages that are supported by the version of JRE that is supplied with Genesys Multimedia). However, Genesys has not tested any language other than those listed above.

FAQ Objects

Taking a category tree and its associated standard responses as input, Knowledge Manager can produce an FAQ object. From this object Knowledge Manager can produce a .jar file, which can in turn be used to:

- Build a web application that accepts written requests and, using content analysis, returns a set of standard responses.
- Present the contents (or a selection from the contents) of the standard response library as answers to frequently-asked questions.

An FAQ object combines a category tree, a training object based on the tree, and, optionally, a model built from the training object. The model is required in order to build a web application.

FAQ objects allow you to include in your web application a means of gathering user feedback about the correctness of a returned standard response. The application then uses this feedback to update the confidence rating of that particular standard response. This functionality is exemplified in the FAQ

sample in the Simple Samples that are installed along with Web API Server. For a description of this sample and its source code, see the *Multimedia 7.6 Web API Client Developer's Guide.*

Sample FAQ .jar File

The sample FAQ that is supplied with Knowledge Manager demonstrates the way that an FAQ object can present a question/answer list. This sample is also included in the Web Compound Samples (see the *Multimedia 7.6 Web API Client Developer's Guide).*

The filename of the sample is `faq_example.jar`. The installation places it in a directory called `FAQExample` in the Knowledge Manager home directory (normally `C:\Program Files\GCTI\MCR 7.6\Knowledge Manager`). To use this sample:

1. Use a text editor to open its batch file `unit_test.bat`.
2. Edit the line

   ```
   set JAVAHOME=D:\jdk1.4\bin\java
   ```

   so that it points to the location of Java in your environment.
3. Use the batch file to launch the sample. You should see the window shown in Figure 71.

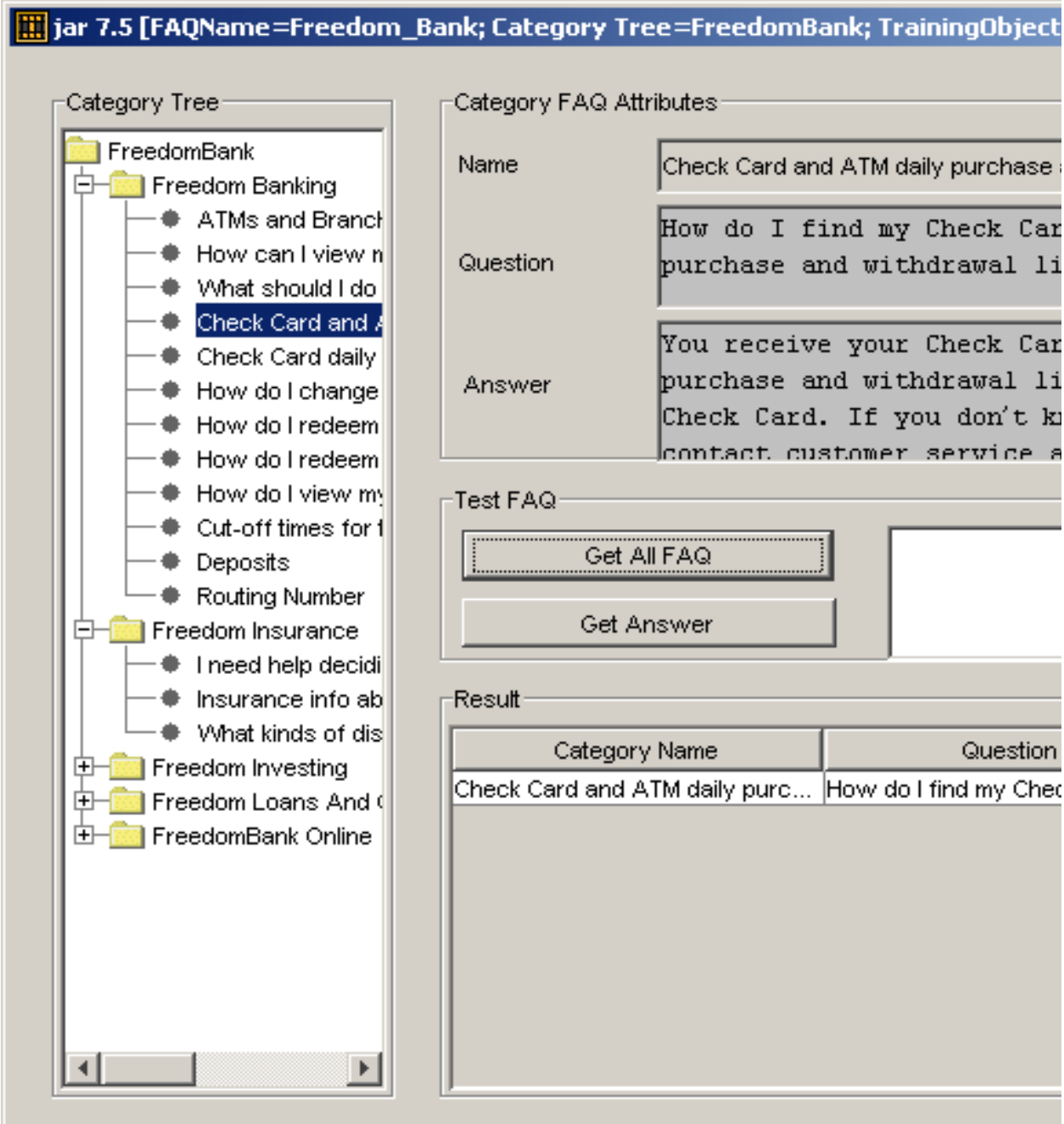


Figure 71: FAQ Sample

This FAQ object works as follows:

- Click a category on the left-hand pane to see its name and associated question and answer displayed in the boxes at upper right.

  The main function of the nonterminal categories (represented as folders) is to organize the terminal categories (represented as grey disks), which are the main locus of questions and answers. Mostly the nonterminal categories do not have answers, and their `Category Question` box displays

either a duplicate of the category name or a short description, rather than a question.

- Click `Get All FAQ` to display, in the `Result` area, a list of all categories contained in the selected category (and its subcategories), along with the question and frequency for each. An example is shown in Figure 72.

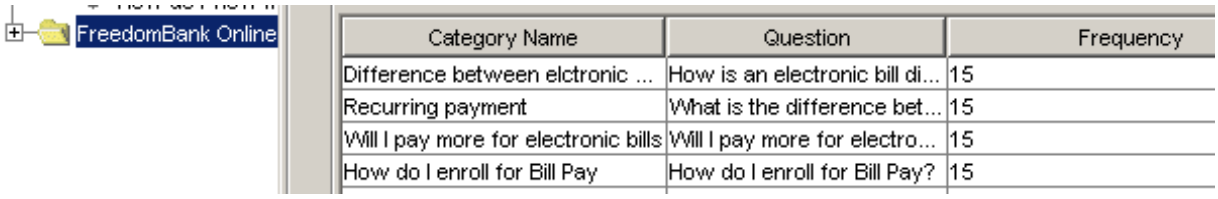


Figure 72: Get All FAQ for a Selected Category

The `Result` area lists all subcategories of the category that is selected on the left-hand pane, providing the following information:

- Category name
- The question associated with the category
- The frequency rating—that is, the number of text objects that are associated with this category in the training object that is part of the FAQ

- To test the FAQ object, enter a sample question in the `Testing Question` box, then click `Get Answer`. The `Result` area displays the answers that the system provides for the sample question at the selected level of the tree. In addition to the category name, question, and frequency, the system also displays a confidence rating. Confidence takes the model that is part of the FAQ object and tells you how good that model is at assigning new questions to this category (see "Reading and Understanding the Ratings" on page 128).

More About FAQ Objects

To be used in an FAQ object, a category must have all of the following attributes:

- Answer: It must have a standard response of the `FAQ` usage type. This attribute is optional for nonterminal categories.
- Question: It must have an associated question, to which the standard response can serve as the answer.
- Selection: It must be selected for inclusion in the FAQ object.

  You must select the categories to include because you cannot assume that all categories in the tree are suitable for use in an FAQ list. For example, your tree might include a category that exists only to provide the standard response *Your account is overdrawn.* Or you might want to use a single category tree to produce multiple FAQ objects, with some categories selected in one FAQ object but not in others.

Requirements are slightly different for terminal categories (those without subcategories) and nonterminal categories (those with subcategories), as shown in Table 10.

Table 10: Required Attributes for FAQ Categories

Attribute	Nonterminal Category	Terminal Category
Answer	Optional	Required
Question	Required	Required
Selected	Required	Required

You can create answers and questions either on the `Categories` tab or the `FAQ` tab. But selection can be done only on the `FAQ` tab, after you generate the FAQ object.

Creating an FAQ Object

Use the following procedure.

Procedure: Creating a new FAQ object

Start of procedure

1. On the `FAQ` tab, do one of the following:
   a. Select `File > New`.
   b. On the left-hand pane, right-click and select `New FAQ Object`.

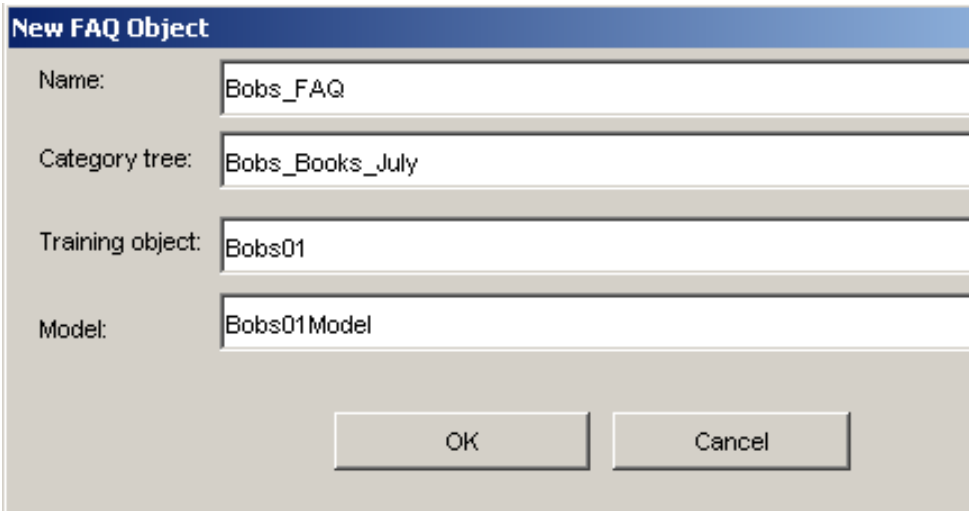


Figure 73: New FAQ Object Dialog Box

2. The `New FAQ Object` dialog box appears, as shown in Figure 73. On it:
   a. Enter a name.
   b. Select a category tree, training object, and model.

   The category tree and training object are required. You can create an FAQ object without a model, but you will not be able to use it in conjunction with content analysis.
3. Click `OK`.

The `FAQ` tab then appears as in Figure 74.

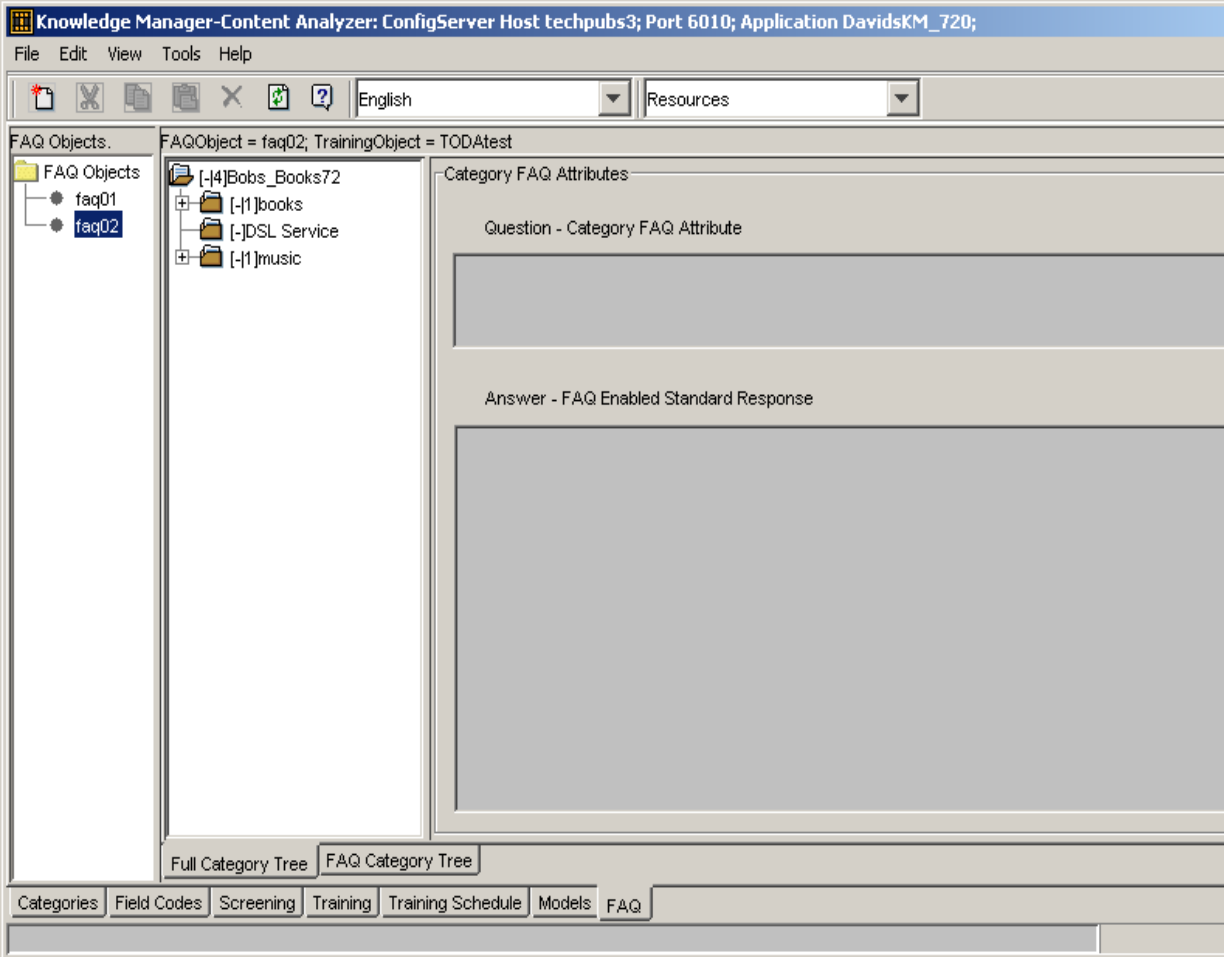


Figure 74: FAQ Object

End of procedure

The `FAQ` tab contains the following panes:

- Left—Displays a list of all FAQ objects
- Center, with two subtabs:
  - Full Category Tree—Displays the entire category tree that serves as the source of the FAQ object that is selected in the left pane
  - FAQ Category Tree—Displays the selected FAQ object itself
- Right—These panes vary, depending on which of the center subtabs is selected.

Full Category Tree Subtab: Configuring the Category Tree

Expanding the category tree on the center pane produces the details shown in Figure 75.

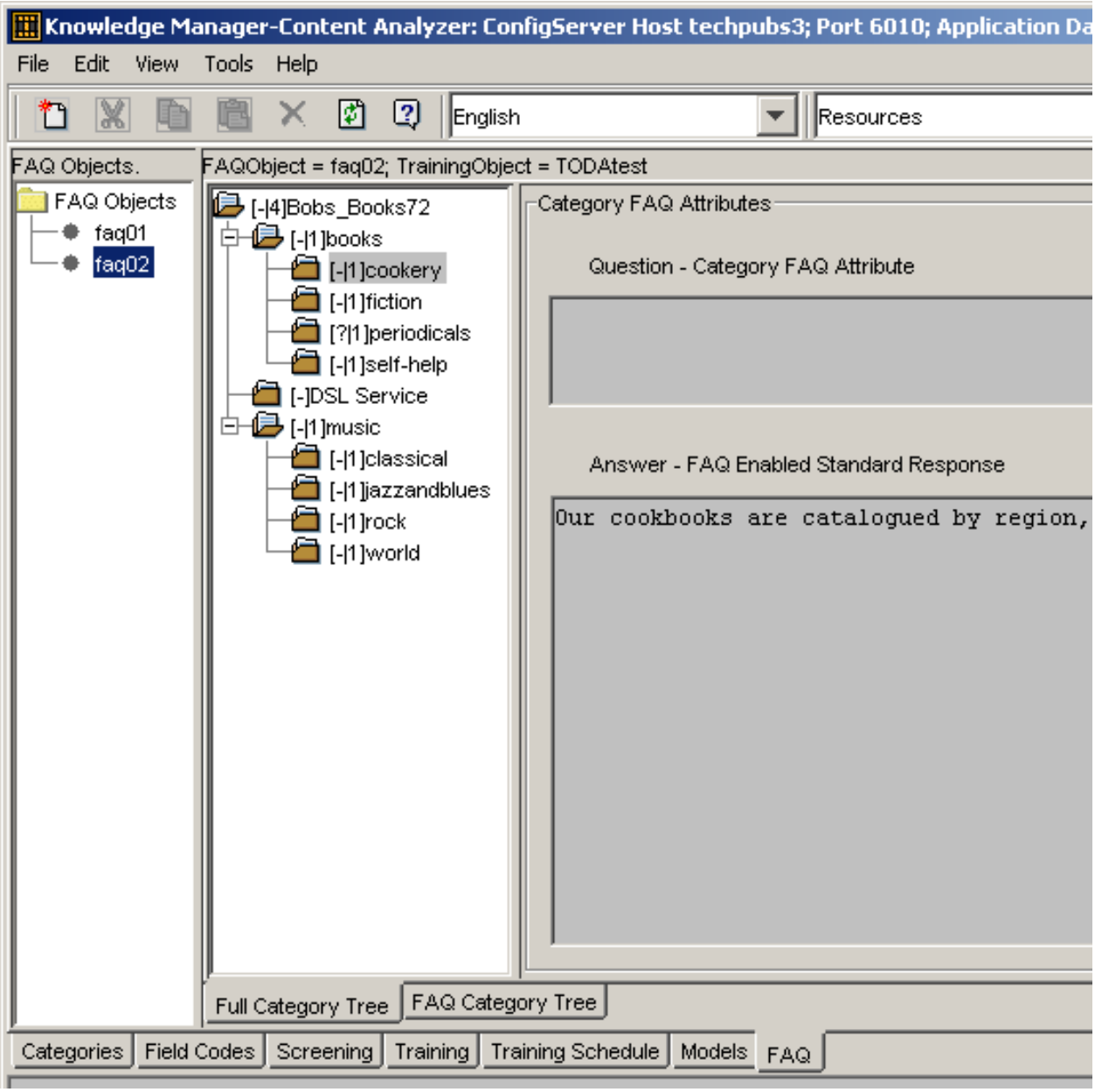


Figure 75: Center Pane of FAQ Object Tab

The following information displays:

- The strip above the center and right-hand panes displays the names of the FAQ object, the training object, and the model if any.
- The right-hand panes display the question and answer (standard response) associated with the category that is selected on the center pane.

The display on the center pane requires more explanation.

On the center pane, each category name appears in the form
`[<statuscode>|<number>]<categoryname>`

<number> is the number of text objects (e-mails and other objects) associated with this category in the training object.

The status code combines the following information:

- Whether the category has the attributes that are required for it to be selected for use in the FAQ object.
- Whether the category is selected for use in the FAQ object.

Table 11 lists the possible statuses:

Table 11: Category Status in FAQ Objects

Status	Meaning
-	Category lacks the required attributes; therefore it cannot be selected.
?	Category has the required attributes but is not selected
+	Category has the required attributes and is selected

Answer

In the context of an FAQ object, an answer is a standard response that has the FAQ usage type selected and is specified as `Active`. You select this usage type on the `Additional` tab of the `New Standard Response` or `Edit Standard Response` dialog boxes. There are two ways to access these dialog boxes:

- On the `Categories` tab, as described in "Creating Standard Responses" on page 28.
- On the `FAQ` tab:
  - To edit an existing standard response, right-click a category and select `Edit Response`.
  - To create a new standard response, right- click a category and select `Set Response`.

Question

Create these questions on the `FAQ attribute` tab of the `New category` or `Category` dialog boxes. As with the usage type of a standard response, there are two ways to access these dialog boxes:

- On the `Categories` tab, as described in "Creating a Category Tree" on page 25.
- On the `FAQ` tab, right-click a category and select `Set/Edit Question`.

Selection

To select a category for inclusion in the FAQ object, right-click it and select:

- `Select` to select only this category.
- `Select With Children` to select this category and all subcategories under it.

Most-Used FAQs

Right-clicking anywhere on the `Full Category Tree` subtab and selecting `Highlight Top Categories` opens the `Highlight most-populated categories` dialog box, as shown in Figure 76.

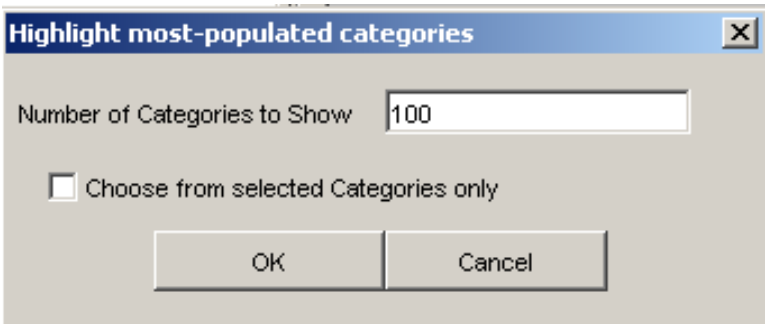


Figure 76: Highlight Most-Populated Categories Dialog Box

When you click `OK`, the dialog box closes, and the center pane highlights, in red, the top *X* categories, where:

- *X* is the number in the `Number of Categories to Show` box.
- *top* means having the largest number of associated text objects in the training object.

Select the `Choose from selected Categories only` check box if you want the calculation to consider only the categories that you have selected for inclusion in the FAQ object (the ones with `+` status).

Figure 77 shows an example.

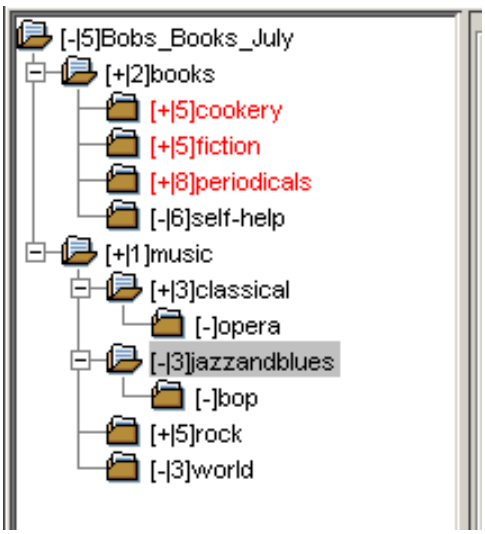


Figure 77: Three Most-Used FAQs

Figure 77 is the result of the following settings:

- `Number of Categories to Show` is set to 3.
- `Choose from selected Categories only` is selected.

If the `Choose from selected Categories only` check box were not selected, the highlighted categories would be `periodicals`, `self-help`, and `Bobs_Books_July`.

Note: If multiple categories are in a tie for inclusion in the top X, the categories that come first (highest) in the listing are the ones highlighted. In Figure 77 there are three categories with five text objects, but the top three has room for only two of them. So `cookery` and `fiction` are highlighted, but not `rock`, which is further down the list.

Shortcut Menu

The complete list of commands that appear when you right-click a category is as follows:

- `Select`—Select the category.
- `Select With Children`—Select the category and all subcategories under it.
- `Unselect`—Unselect the category.
- `Unselect With Children`—Unselect the category and all subcategories under it.
- `Expand Tree`—Self-explanatory.
- `Collapse Tree`—Self-explanatory.
- `Highlight Top Categories`—Open the `Highlight most-populated categories` dialog box.
- `Set/Edit Question`—Create or edit a question for the category.
- `Set Response`—Create a new standard response for the category. For the response to appear in the FAQ object, you must go to the `Additional` tab and select `Active` FAQ usage.
- `Edit Response`—Edit an existing standard response for the category.

FAQ Category Tree Subtab: Viewing and Testing

After you have finished configuring the category tree, you can display the result on the `FAQ Category Tree` subtab, as shown in Figure 78.

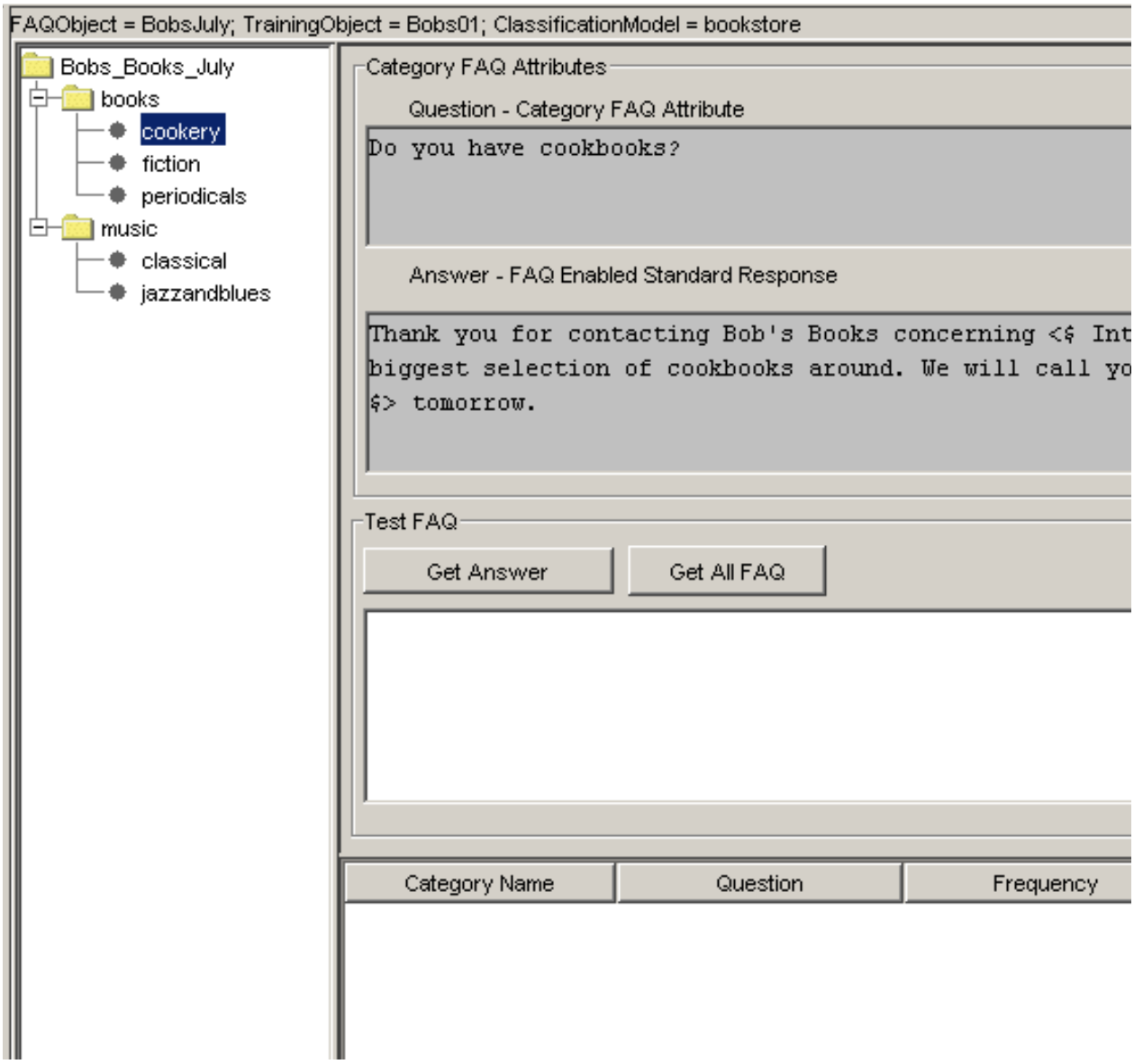


Figure 78: FAQ Category Tree Tab

The FAQ Category tree is a preview of the FAQ object using a format similar to the sample described previously (see "Sample FAQ .jar File" on page 143).

Selecting a category on the `FAQ Category Tree` subtab produces the following results:

- The `Category FAQ Attributes` area displays the question and answer for the category that is selected on the center pane.
- In the `Test FAQ` area, click `Get All FAQ` to display a list of all categories contained in the selected category (and its subcategories), along with the question and frequency for each

Notes: Text in the text box has no effect on this action.

Categories that have questions but no answers do not appear on this list.

- In the `Test FAQ` area, enter a sample question in the text box, then click `Get Answer` to display the answers that the system provides for the sample question at the selected level of the tree. (If the FAQ object does not include a model, this button is dimmed.) An example is shown in Figure 79.

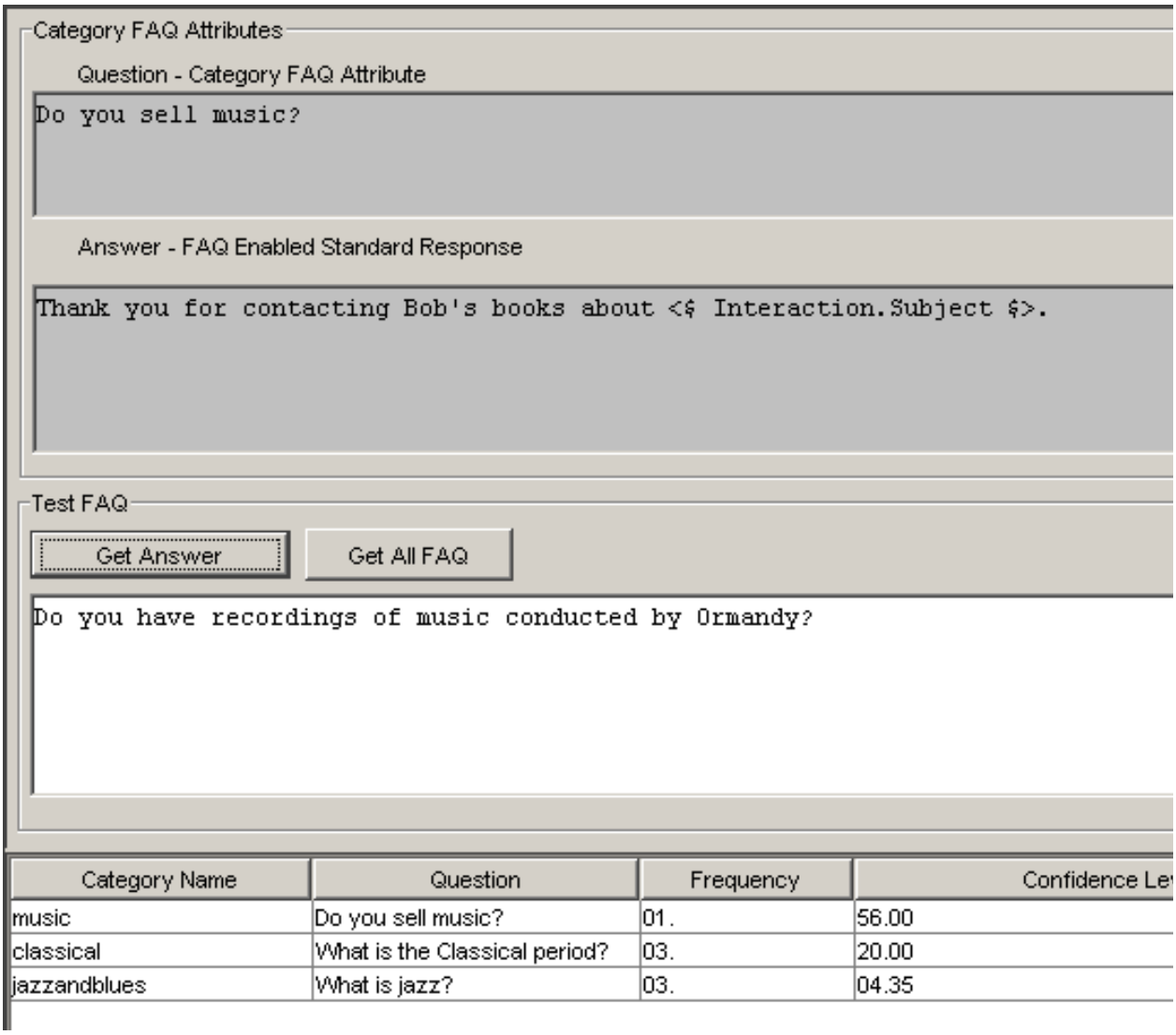


Figure 79: Test FAQ: Get Answer

Figure 79 shows the result of the question *Do have recordings of dance music conducted by Ormandy?* with the `Music` category selected (see Figure 78 for the category tree in question).

The bottom pane lists all answers supplied by the system, showing the category name, question, frequency (meaning the number of text objects associated with the category in the training object, also shown as <number> on the `Full Category Tree` tab display), and confidence rating (see "Reading and Understanding the Ratings" on page 128).

Generating an FAQ.jar File

Use the following procedure.

Procedure: Generating and testing an FAQ.jar file

1. Select an FAQ object on the left-hand pane, then do one of the following:
   - Select `Tools > Build FAQ JAR file.`
   - Right-click on the left-hand pane and select `Build FAQ JAR file.`

The `Build FAQ Jar file` dialog box appears, as shown in Figure 80.

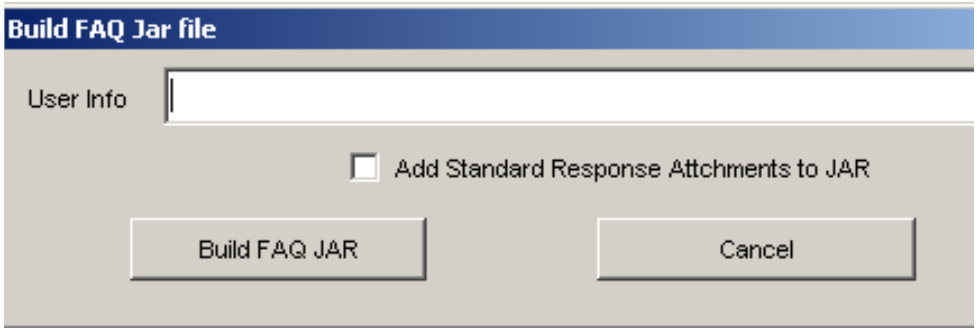


Figure 80: Build FAQ Jar File Dialog Box

2. In the `User Info` text box, enter the text that you want to appear in the title bar of the window that displays the FAQ object when you test it (see the procedure that immediately follows).
3. Select `Add Standard Response Attachments to JAR` if the answers in your FAQ object have attachments that you want to include in the FAQ.jar file.
4. Click `Build FAQ JAR.`
5. The `JAR File Chooser` dialog box appears. Enter a file name.
6. To test the resulting `FAQ.jar` file, make a copy of the `unit_test.bat` file that accompanies the FAQ example.
7. Edit it so that it targets the .jar file that you built instead of `faq_example.jar.`

   Running the .bat file displays your FAQ object in the same way as the FAQ example (see "Sample FAQ .jar File" on page 143). Notice that the title bar

of the FAQ object window displays the text that you entered in the `User Info` text box of the `Build FAQ Jar file` dialog box.

End of procedure

Next Steps

- You can now use the `FAQ.jar` file to create web applications.

Typical Response Times

This section describes some typical response times for Genesys Content Analyzer. For other functions of Knowledge Manager see "Typical Response Times" on page 82 of Chapter 2.

Unless otherwise stated, these figures are for a machine running Windows 2000 with two Pentium 4 processors and 1 GB of RAM.

- Deleting a training object takes approximately 4 seconds per 1,000 e-mails.
- Copying e-mails from one training object to another takes approximately 8 seconds per 1,000 e-mails.
- Creating a model (training time) naturally varies with the number of categories, number of e-mails, selected training quality, and selected cross-validation. As one example, for a training object containing 76 categories and 73,000 mails, with training quality set to level 1 and no cross-validation, training time is approximately 29 minutes. This is on a host running Windows 2000 with one 600 MHz processor and 1 GB of RAM.
- Cross-validation may increase training time significantly. Table 12 shows, for selected cross-validation levels, the factors of increase of cross-validation over no cross-validation.

Table 12: Increase of Training Time with Cross-Validation

Cross-Validation Level	Factor
3	1.9–2.9
5	4.0–4.7
10	8.0–9.0

For example, training a model at cross-validation level 3 takes between 1.9 and 2.9 times as long as the same model with no cross-validation.

- Classification performance depends on the size and nature (level of training quality) of the model. Table 13 shows some examples, all of which use a test object that contains 3,726 text objects.

Table 13: Classification Performance

Model	Host Machine	Classification Rate
• 72,734 text objects • Size = 285 KB • Quality = 1 • Cross validation with split to three sets	• Two Pentium 3 processors • 512 MB RAM • Windows operating system	31 objects classified per second
• 72,734 text objects • Size = 309 KB • Quality = 3 • Cross validation with split to 10 sets	• Two Pentium 3 processors • 1 GB RAM • Windows operating system	28 objects classified per second
• 72,734 text objects • Size = 309 KB • Quality = 3 • Cross-validation with split to 10 sets	• Four 350 MHz processors • 4 GB RAM • Solaris operating system	15 objects classified per second

- An FAQ object can process 30–50 classification requests per second on the model that contains 500–1,000 categories.

Chapter

3 Multi-Tenancy

This chapter describes multi-tenant configuration of Multimedia. It covers these topics:

- Overview, page 159
- Configuration, page 159
- Limitations, page 160

Overview

Multi-tenancy is the capability of maintaining a pool of resources and controlling access to it by more than one tenant or business instance. For general information about multi-tenancy, see the *Framework 7 Configuration Manager Help* and the *Framework 7 Getting Started Guide.*

Configuration

The following Multimedia applications can be shared across tenants: Chat Server, Interaction Server, Universal Contact Server (UCS) and Web API Server. For all other Multimedia applications you must deploy one instance per tenant.

In a multi-tenant environment, each Multimedia application must have one Tenant specified on its `Tenants` tab, with the following exceptions:

- UCS, Chat Server, and Interaction Server can have more than one Tenant specified.
- Web API Server can have more than one Tenant specified if the server itself is an object of load balancing.

This means that clients of Web API Server must include a `tenant` parameter in their requests, even in single-tenant environments, and even if the client itself is single-tenant.

Note: Changes to Interaction Server's tenant specification do not take place dynamically—you must restart Interaction Server for changes to take effect.

Interaction Server

Since release 7.2, Interaction Server has had two possible Application types, `Interaction Server` and `T-Server`. With type `Interaction Server` it does not require an associated multimedia switch. Clients of Interaction Server should not expect it to specify any switch.

Note: Support of multi-tenancy requires the use of the `Interaction Server` application type.

For backward compatibility with release 7.1, clients are nevertheless able to associate Interaction Server with a multimedia switch, as follows: If a Tenant that is specified by Interaction Server contains a multimedia switch, clients associate Interaction Server with this switch. This works only if the Tenant contains exactly one multimedia switch.

See also "Interaction Server" on page 174 for other consequences of this difference in application type.

Limitations

Observe the following limitations:

- Deploy at most one multimedia switch per tenant.
- Deploy at most one Interaction Server per tenant.
- Deploy at most one UCS per tenant.
- Deploy at most one UCS database per tenant. Databases can be shared between tenants. Figure 81 shows possible ways of arranging multiple tenants, Universal Contact Servers, and databases.

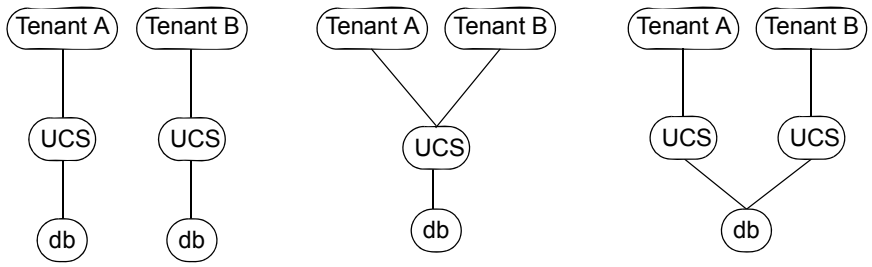


Figure 81: Multi-Tenant Configurations with UCS and the UCS Database

Chapter

4 Load Balancing and Backup Configuration

This chapter describes load balancing. It covers these topics:

- Overview, page 161
- Web API Server, page 162
- Interaction Server, page 165
- Backup Configuration, page 167

Overview

Load balancing provides greater scalability and availability of service by providing multiple instances of certain Multimedia servers. The redundancy provided by load balancing helps prevent loss of data.

Load balancing can take place within a tenant or across tenants. There are three types of load balancing:

- Web API Server can balance among multiple instances of the following servers within a single tenant or across tenants:
  - Callback Server
  - Chat Server
  - Co-Browsing Server
  - E-mail Server Java
  - Interaction Server
  - Stat Server
  - UCS

  Web API Server does this using the Load-Balancing API. It keeps track of available server instances through Solution Control Server (SCS).
- Web API Server can balance among multiple instances of its own type (multiple Web API Servers).

- Interaction Server can balance, within a single tenant only, among multiple instances of Classification Server and E-mail Server Java. It can also balance among multiple instances of Universal Routing Server (URS) as long as all instances have the same strategy loaded. See “Interaction Server” on page 165 for details.

Web API Server

Web API Server uses SCS to monitor the state of all servers. Load balancing that involves Web API Server includes:

- Load balancing between instances of Web API Server.
- Load balancing between instances of the following servers: Callback Server, Chat Server, E-mail Server Java, Interaction Server, Stat Server, UCS.

Load-balancing components react to:

- Changes in the configuration; specifically, the connection settings of Web API Server and the status (enabled or disabled) of relevant applications.
- Application states (running or stopped) as reported by SCS.

Load balancing performs the following actions:

- For all instances of servers that are listed on Web API Server’s `Connections` tab:
  - **a.** The instance reports its status as RUNNING to SCS when it starts. The instance is then ready to process interactions.
  - **b.** The load balancer then marks the server’s status as RUNNING.
  - **c.** A web application can now use the service provided by that instance.
- If the server’s status changes to Service Unavailable:
  - It is excluded from the list of available servers and cannot be given in response to a web application’s request.
  - If the server’s status changes again to Started (Service Available), it returns to the list of available servers. In the case of Chat Server, it can continue to handle an online session that was established before the status changed to Service Unavailable.
- If the server shuts down:
  - It is excluded from the list of available servers and cannot be given in response to a web application’s request.
  - All interactions that it was handling are either lost (in the case of Chat Server) or handled by other instances of the same type of server (E-mail Server Java).
- If the server is disabled in the Configuration Layer:
  - It is likewise excluded from the list of available servers and cannot be given in response to a web application’s request.

  - None of the clients that are already working with this instance of the server are affected.
  - As soon as all interactions that the server is handling end, the server can be shut down.
- If the server is enabled, it returns to the list of available servers.

Load-Balancing Configuration for Web API Server

In the example Multimedia configuration shown in Figure 82 on page 163, there are two instances of Web API Server, two of E-mail Server Java, and three of Chat Server, with the Chat Server instances divided between two tenants. All application servers are available to both Web API Servers.

Note: To enable cross-Tenant load balancing, you must add the Tenants to the `Tenants` tab of the Web API Server that you want to serve as the point of load balancing.

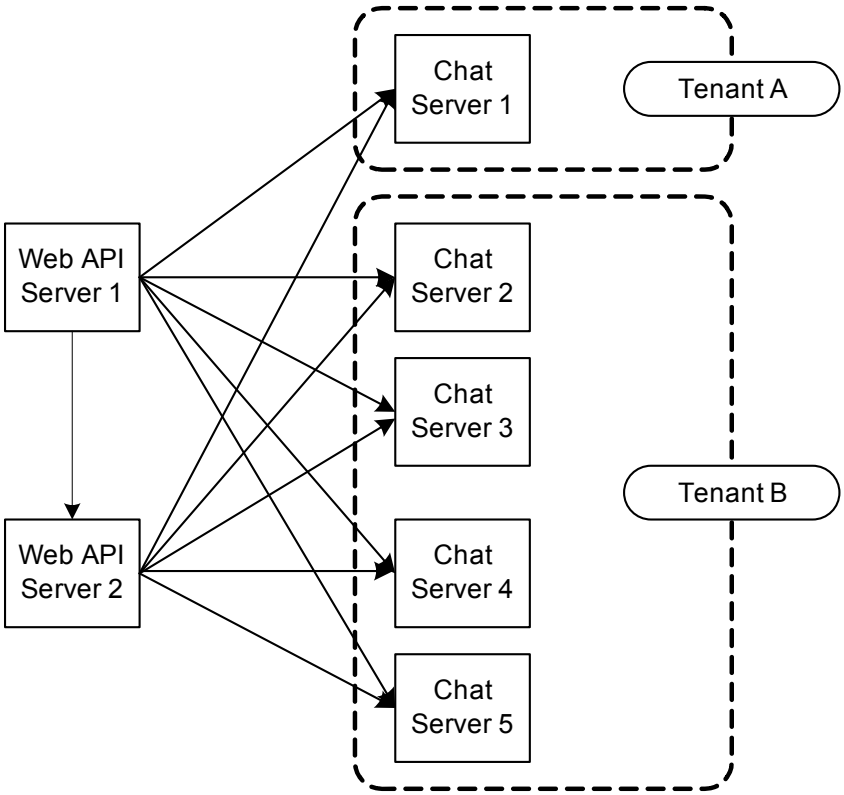


Figure 82: Load Balancing Configuration

To simplify the overall configuration, you can use application objects of type `Application Cluster` to group available servers in the configuration. An Application Cluster is a configuration object that stores connection specifications. The Multimedia configuration wizard offers the opportunity to

create Application Clusters. If you did not create an Application Cluster while running the wizard, you can add one manually, using the following procedure.

Procedure: Configuring an Application Cluster

Start of procedure

1. In Configuration Manager, locate the template `ApplicationCluster_<version-number>.apd` in the `templates` directory of your product CD and import it.
2. Use the template to create a new Application Cluster.
3. On the `Connections` tab, add connections to the servers that Web API Server will balance.
4. On the `Server` and `Start Info` tabs, enter arbitrary characters in the empty fields. Web API Server ignores this information, but there must be something in these fields for you to be able to save the Application.
5. In your Web API Server Application object or objects, add a connection to the Application Cluster.

End of procedure

Application Clusters are transparent to load-balancing components, and the configuration that uses connections through Application Clusters is equivalent to a configuration that uses direct connections between servers.

The configuration shown in Figure 82 is reconfigured in Figure 83 using an Application Cluster.

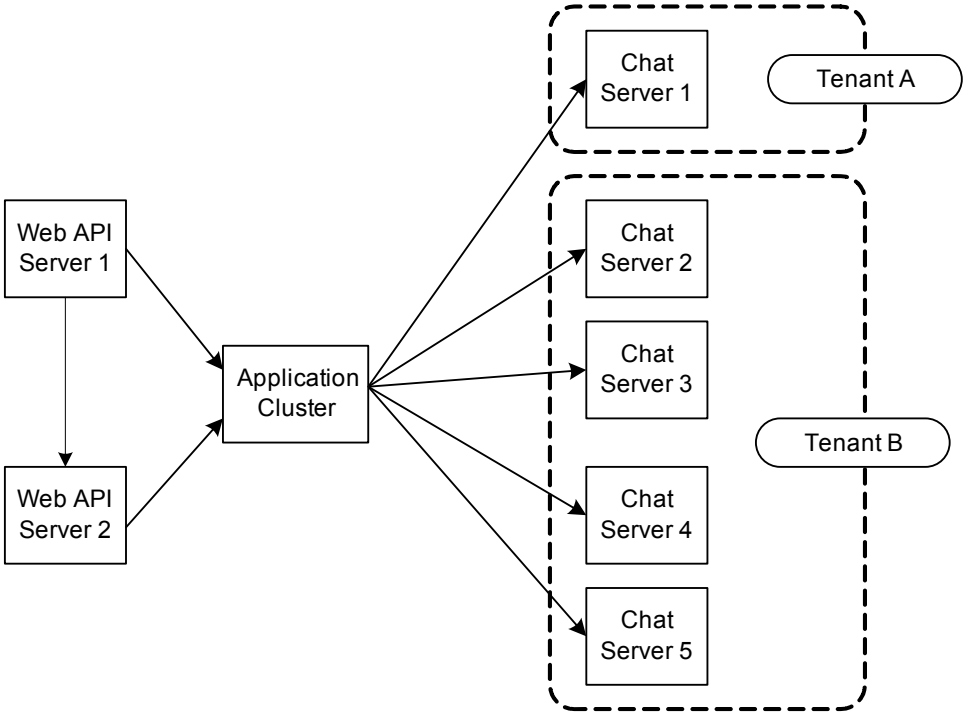


Figure 83: Load Balancing with Application Cluster

Load-Balancing API

To use Multimedia 7.6 load-balancing capabilities, use the Load-Balancing API in the web application.

The Load-Balancing API provides the following functionality:

- It can select a particular server instance from a set of instances of the specified server type.
- Upon the first request to a server instance, it can create an alias for the selected server instance and store it for future use.
- It can use the alias to obtain connection parameters (host name and port) of the server instance.
- It has access to configuration information.

The Multimedia 7.6 simple samples demonstrate how to use the Multimedia Load-Balancing API to develop web applications that use load balancing. For details, see the "Using Load-Balancing API" chapter in the *Multimedia 7.6 Web API Client Developer's Guide.*

Interaction Server

Interaction Server can balance among multiple Universal Routing Servers and among Multimedia application servers. It can also connect to its database via

multiple Database Access Points (DAPs). It does not use a user-accessible API for load balancing.

Balancing Universal Routing Servers

Interaction Server can balance among multiple instances of URS. This balancing proceeds by strategy: when an interaction reaches a strategy object in a workflow, Interaction Server selects (in round-robin fashion) from among all URS instances that have that strategy loaded.

To enable this type of load balancing, you must:

- Configure a connection from each URS to Interaction Server.
- For all participating URS instances, set the `agent_reservation` option to `true`.

> **Note:** For additional information on the `agent_reservation` option, see Chapter 5 in *Universal Routing 7.6 Reference Manual*.

- Choose each URS when activating each strategy in Interaction Routing Designer (IRD). Do this by shift-clicking all of the desired URS instances in the `Choose Routing Server` window of the Strategy Activation Wizard.

Suppose Interaction Server has two URS instances connected to it: URS 1 has Strategies A and C loaded, and URS 2 has Strategies B and C loaded. Then,

- For interactions that arrive at Strategy A in a workflow, Interaction Server submits them to URS 1.
- For interactions that arrive at Strategy B in a workflow, Interaction Server submits them to URS 2.
- For interactions that arrive at Strategy C in a workflow, Interaction Server balances between URS 1 and URS 2.

If any instance of URS shuts down, Interaction Server detects that this instance is not available. If any interactions were pending in the unavailable URS, Interaction Server resubmits them to an available URS that has the required strategies loaded.

Balancing Multimedia Application Servers

An *application server* is a server that Interaction Server invokes when triggered to do so by a routing strategy. For example, a Classify object in a strategy triggers Interaction Server to invoke Classification Server. To do this, Interaction Server uses a protocol called *External Services Protocol* or ESP; therefore these servers are also called ESP servers. The application servers that Interaction Server can balance among are:

- Classification Server.
- E-mail Server Java.

To enable this type of load balancing, you must configure connections from Interaction Server to each instance of the application server.

Note: E-mail Server Java has a dual role, as follows:

- When Interaction Server contacts it using ESP (for example, asking it to generate an autoresponse), E-mail Server Java is an application server (ESP server) and Interaction Server is its client.
- When E-mail Server contacts Interaction Server, using the Interaction Management Protocol, and asks to submit an incoming e-mail interaction, it is a media server and a client of Interaction Server.

  For more information on these protocols, see the *Genesys 7 Events and Models Reference Manual.*

Balancing DB Servers

Interaction Server must work with only one database. However, Interaction Server supports multiple DAP connections to the same database through different DB Servers. You can configure this using multiple connections (DAPs) to one database.

Note: If for some reason the configuration of Interaction Server's DAP is erroneously changed to point to a different database, Interaction Server overlooks the error: it sends an error message saying that the configuration has changed, and continues to work with the original database. But if, in this same scenario, a switchover to a backup Interaction Server occurs, the backup Interaction Server has no way of knowing that the DAP configuration has (erroneously) changed, so it connects to the new database and starts sending requests to it.

It must be emphasized that Genesys recommends against that you do not change the DAP configuration on the fly.

Backup Configuration

This section describes the types of backup configuration supported in Multimedia 7.6

High-Availability Support

Table 14 lists the types of high-availability support possible for Multimedia servers.

Table 14: High-Availability Support

Component	Type of Support
Chat Server	Load balancing[a]
Classification Server	Load balancing[b]
Co-Browsing Server	Load balancing[a]
E-mail Server Java	Warm standby and load balancing[a,b]
Interaction Server	Warm standby
Interaction Server Proxy	Warm standby
Training Server	Load balancing[c]
Universal Contact Server	Warm standby
Universal Contact Server Proxy	Warm standby
Web API Server	Load balancing[a]

a. Supported through load balancing on Web API Server.

b. Supported through ESP load balancing (see page 166) by Interaction Server.

c. Supported in that it can process multiple training jobs. However, if an instance of Training Server becomes unavailable while it is processing a job, then a second running instance of Training Server will not pick up the job for processing. Instead, you must restart the first instance.

For general information on warm standby, see the *Framework 7.6 Architecture Help* and the *Framework 7.6 Deployment Guide.*

Chapter

5 Ongoing Administration and Other Topics

This chapter describes ongoing administration, security, and other topics in these sections:

- Administration, page 169
- Security, page 175
- Limitations, page 176
- E-mail Server Java: Advanced Topics, page 182
- Interaction Server: Advanced Topics, page 188

Administration

This section presents some recommendations for monitoring and adjusting your Multimedia configuration.

General Recommendations

Parameters to Check

Check that the following parameters do not significantly exceed their average values:

- Memory usage
- CPU load
- Number of handles for Multimedia-related processes (with Windows operating system)

Loading on Application Servers

Monitor the loading on application servers (Classification Server, E-mail Server Java). If application servers are being overloaded, do one or both of the following:

- For all routing strategies that process interactions with no agent involvement, adjust the limit on the number of interactions that Interaction Server can submit to Universal Routing Server (URS). You can set this limit for a strategy using the `max-submitted-interactions` option. See "Interaction Server Options" in Chapter 2 of the *Multimedia 7.6 Reference Manual.*
- Add instances of the required application server on other hosts.

Database Performance

If you are running Microsoft SQL, Genesys recommends as follows.

Microsoft SQL 2000

- In general, patch up to Service Pack 4.
- If Microsoft SQL is running on a machine with over 2 GB of RAM, use Windows' AWE (Address Windowing Extensions) mode. To avoid performance degradation, patch Microsoft SQL according to Microsoft's recommendation "FIX: Not all memory is available when AWE is enabled on a computer that is running a 32-bit version of SQL Server 2000 SP4" (see http://support.microsoft.com/?kbid=899761). This patch brings Microsoft SQL to version 8.00.2040.

Microsoft SQL 2005

- There is an issue that occurs with Microsoft SQL 2005: when the database is very large (on the order of one million interactions), there are periodically exceptions in the `Stat` service, and CPU activity rises to 100%.

  To avoid this issue, configure the UCS DAP as follows:

  a. Create a `settings` section.

  b. In this section, create an option called `prepare` and set its value to `false`.

  This DAP configuration applies to Microsoft SQL 2005 only; configuring the DAP in this way with Microsoft SQL 2000 degrades performance.

UCS

Character Sets

The character set `WE8ISO8859P1` does not have any representation of characters in the range 128–159. Because of this, with an Oracle database, attempting to save characters in this range in a column of type `NCHAR` or `NVARCHAR` results in corrupted data. Genesys recommends that you set the Oracle `NLS_CHARACTERSET` parameter to `WE8MSWIN1252` instead of `WE8ISO8859P1`. `WE8MSWIN1252` is a superset of `WE8ISO8859P1`, so there will be no data loss.

For support of nonlatin charsets, use the following parameter settings in Oracle:

```
NLS_CHARACTERSET AL32UTF8
NLS_NCHAR_CHARACTERSET AL16UTF16
```

Access to Configuration Server

Be sure to run UCS with a user that has write access to the Configuration Server database for all the tenants associated with this UCS (that is, the user specified on the `Security` tab of the UCS Application object).

This means that UCS does not support Configuration Server Proxy, which has only read access to the Configuration Server database.

Contact Identification and Creation

If UCS cannot identify a contact, its default behavior is to create a new contact record. For description of this behavior and ways to customize it, see Chapter 6, “Contact Identification and Creation,” on page 197.

Database Tuning for Attachments

UCS uses the `Content` field of the `Document` table to store attachments; also, the `Content` field of the `Interaction` table stores raw e-mails, including attachment data. If you plan to store large attachments (bigger than 5 MB), you should tune the database according to the recommendations of your database vendor.

For example, increasing the block size of database files for these fields can greatly enhance performance in access and storing of large attachments, at the cost of a slight loss of performance with smaller ones. Also, some databases offer the ability to partition data according to specified criteria. Both tables have a `theSize` column that you can use to do such partitioning. This could enable you to store small attachments in a specific file and large ones in another, for example.

Refer to the tuning guides of your database vendor for more information.

User Other than Schema Owner

To enable a user who is not the schema owner to run UCS with an Oracle database:

1. Open the script `ucs_oracle_create_additional_user.sql`, which is located in the `sql_scripts` directory of UCS's starting directory.
2. Locate the following lines:

   ```
   ucs_user := 'UCS_RUNTIME';
   ucs_db_creator := 'UCS_OWNER';
   ```

3. Replace `UCS_RUNTIME` with the name of the non-owner that you want to be able to run the database.
4. Replace `UCS_OWNER` with the name of the Oracle user that created (is the owner of) the schema.
5. Run the script from an Oracle account that has SYSDBA privileges. This creates the user identified in Step 2 and creates synonyms of all objects so that they are accessible to the newly created user.
6. Adjust the DAP that UCS uses (or create a new DAP if you want to retain the existing one), as follows:
   - On the DB Info tab, set the `User Name` equal to the user identified in Step 2, and set the password equal to the user name. This is how the script creates the user and password. If you want a different password you must modify it in Oracle.
   - On the `Options` tab, `settings` section, create the `db-schema-name` option. For its value, enter an upper-case version of the name of the Oracle user that created the UCS database schema (the user identified in Step 3).
7. After completing these steps on the main database, repeat them for the archive database.

Note: When using UCS with a limited DB user, UCS is not able to check for the existence of table indexes. The log will display a message warning that indexes do not exist, but you can ignore this warning. Using a limited user does not prevent DB from using indexes.

Required Queries

Users of the UCS DB must have permission to run certain queries on the database, for the following purposes:

- List user's tables—required for launching UCS
- Read NLS (national language support) configuration—required for normal operation of UCS
- List user's indexes—required for normal operation of UCS

- Read the configured maximum number of cursors—required for normal operation of UCS, on Oracle only

The following examples use `CONTACTSERV_USER` and `CONTACTSERVARC_USER` as the names of users of the main and archive databases respectively:

Oracle

To list user's tables:

```
SELECT T.TABLE_NAME, T.COLUMN_NAME, DECODE (T.NULLABLE, 'N', 'NO',
'YES') AS IS_NULLABLE FROM ALL_TAB_COLUMNS T WHERE
UPPER(T.OWNER)='CONTACTSERV_USER' ORDER BY T.COLUMN_ID
```

To read NLS configuration:

```
SELECT * FROM sys.props$ WHERE name LIKE 'NLS%CHARACTERSET%'
```

To list user's indexes:

```
SELECT I.INDEX_NAME, IND.COLUMN_NAME as COLUMN_NAME, I.UNIQUENESS as
IS_UNIQUE FROM USER_INDEXES I, USER_IND_COLUMNS IND WHERE
I.INDEX_NAME = IND.INDEX_NAME AND I.GENERATED='N' ORDER BY
INDEX_NAME
```

To read the configured maximum number of cursors:

```
SELECT value FROM v$parameter WHERE name = 'open_cursors'
```

SQLServer

To list user's tables:

```
exec sp_tables  @table_name = null,  @table_type = '''TABLE'''
```

To read NLS configuration:

```
SELECT DATABASEPROPERTYEX ('CONTACTSERV_USER', 'Collation')
```

To List user's indexes:

```
SELECT i.name INDEX_NAME, c.name AS COLUMN_NAME, CASE WHEN (i.status
& 2)<>0 THEN 'true' ELSE 'false' END AS IS_UNIQUE FROM sysindexes i
INNER JOIN sysindexkeys k ON i.id=k.id AND i.indid=k.indid INNER
JOIN syscolumns c ON c.id=i.id AND c.colid=k.colid WHERE
INDEXPROPERTY (i.id , i.name , 'IsAutoStatistics' ) = 0 ORDER BY
index_name
```

DB2

To list user's tables:

```
SELECT TABNAME AS TABLE_NAME, COLNAME AS COLUMN_NAME, TYPENAME AS
TYPE_NAME, LENGTH AS TYPE_LENGTH FROM syscat.columns WHERE
UPPER(TABSCHEMA)='CONTACTSERV_USER' ORDER BY COLNO
```

To read NLS configuration:

```
SELECT TYPENAME, CODEPAGE FROM syscat.datatypes WHERE TYPENAME LIKE
'%CHAR%' OR TYPENAME LIKE '%LOB%'
```

To List user's indexes:

```
SELECT INDNAME AS INDEX_NAME, COLNAMES AS COLUMN_NAME, UNIQUERULE AS
IS_UNIQUE FROM syscat.indexes WHERE
UPPER(TABSCHEMA)='CONTACTSERV_USER' ORDER BY INDNAME
```

Interaction Server

Use CCPulse to monitor interaction queues (in interaction workflows) for signs of problems with routing strategies. If the number of interactions in a queue increases abnormally, it may be a sign that the strategy that processes interactions from that queue is not loaded in Universal Routing Server.

As described in "Interaction Server" on page 160, Interaction Server has two possible Application types in the Configuration Layer. `Interaction Server` is the normal type; the `T-Server` type is also available for backward compatibility. Be aware that an Interaction Server 7.6 of type `T-Server`, upon startup, will make two attempts to connect to Configuration Server. The first attempt will generate `trace`-level alarms (about a missing application of type: Interaction Server) that you should ignore. The second attempt will succeed.

Classification Server

Modifying any of the following may have repercussions elsewhere in the system:

- Categories
- Standard responses
- Field codes
- Screening rules

If you modify any of these objects, it would be prudent to check any compiled strategies that use the following:

- Acknowledgment
- Attach Categories
- Autoresponse
- Chat Transcript
- Classify
- Classify Switch
- CreateEmailOut
- CreateSMS
- Forward
- Multi Screen
- Screen

Perform this check by recompiling the strategies in question. If this is not possible, monitor the Classification Server log for errors related to screening rules and UCS logs for errors related to rendering of standard responses.

Web Compound Samples

With the AIX/DB2/WebSphere platform, in a multitenant configuration using a non-English environment, the Web Compound Samples require some modifications to the `constants.jsp` file, as follows.

1. Remove the following line:
   ```
   <%@ page import="Genesys.webapi.utils.i18n.*" %>
   ```
2. Set the correct charset for the language of the environment. For example, if the language is Korean, use the charset EUC-KR and set it as follows:
   ```
   <%@ page contentType = "text/html; charset=EUC-KR" %>
   ```
3. Add the correct tenant name:
   ```
   String strTenant  =  "<tenant_name>";
   ```

You must also set the tenant value for the appropriate options in the Web Client application.

Security

Genesys makes the following security recommendations for deploying Multimedia:

- Put Web API Server in the DMZ.
- Put all other Multimedia components in the internal network.
- Open ports in the firewall between the DMZ and the internal network to allow Web API Server to connect with other Multimedia components. Table 15 lists each component and the port to open.

Table 15: Port Types in Firewall

Server	Port
Configuration Server	Default port on `Server Info` tab
Message Server	Default port on `Server Info` tab
Solution Control Server.	Default port on `Server Info` tab
Interaction Server	Default port on `Server Info` tab
Chat Server	Port specified by the `webapi-port` option in the `settings` section. If not specified, default port on `Server Info` tab.

Table 15: Port Types in Firewall (Continued)

Server	Port
E-mail Server Java	Port specified by the `webapi-port` option in the `settings` section
Stat Server	Default port on `Server Info` tab
Co-Browsing Server	HTTPS
UCS	Port specified by the `ucsapi` option in the `ports` section

- Open a port in the firewall to allow Solution Control Server to connect to the Local Control Agent (LCA) located on the host of Web API Server.

Limitations

This section describes recommended limitations.

General

If you set up periodic time synchronization on your network, be sure to use a dedicated Network Time Protocol (NTP) server. Without NTP, synchronization can cause errors in E-mail Server Java and UCS.

Knowledge Manager

Basic Limitations

For Knowledge Manager, observe the following limitations:

- Categories: 3,500 categories
- Standard responses: 50 per category
- Attachments: 20 per standard response, 5 MB per attachment
- Field codes: 1,000
- Screening rules: 1,000
- Training objects: 200,000 e-mails, 20 KB per e-mail, 510 B for each e-mail's subject field

Screen Resolution

For Knowledge Manager to operate correctly, you must set a minimum screen resolution of 1280 x 1020.

Memory Allocation

You can adjust the memory size that Java allocates for Knowledge Manager processes by using the parameter `-Xmx1000m` in the following line in the .bat file:

```
start "Knowledge Manager" /b "%GES_HOME%\jre\bin\javaw" -Xmx1000m
-classpath %CLASSPATH% -Djava.security.manager
-Djava.security.policy=.\java.policy Genesys.iknow.manager.TM_start %*
```

`-Xmx1000m` means that 1,000 MB is allocated for Knowledge Manager; changing this number changes the allocation. The following considerations bear on adjusting this parameter.

- In some cases, Knowledge Manager does not work when you attempt to launch it from a machine that has a remote connection to the host of Knowledge Manager. As a workaround, lower the value of `-Xmx1000m` to `-Xmx512m`. In the unlikely event that this does not work, try a further decrease to `-Xmx256m`.
- You may want to adjust this parameter for better performance with large training objects (see "Large Training Objects" on page 118), or before importing or exporting large files. For DB2 and Oracle, see also the recommendations in "Adjusting Database Configuration" below.

However, if this parameter is too low, it may impose limits on Knowledge Manager lower than those listed in "Basic Limitations" above. If so, you can consider increasing this parameter.

Adjusting Database Configuration

To prevent problems when using Knowledge Manager to import or export very large files, Genesys has the following recommendations about database configuration.

- For DB2, do as follows:
  - **a.** In the DB2 Control Center, select `System > Instance > Databases`.
  - **b.** Select the database desired.
  - **c.** Right-click on the desired database.
  - **d.** In the resulting shortcut menu, select `Configure`.
  - **e.** In the resulting dialog box, select `Logging`.
  - **f.** Increase the number of files and/or file size.
- For Oracle, use Enterprise Manager to increase the number of rollback segments. Refer to Oracle documentation for details.
- For Microsoft SQL, no special configuration is needed.

Chat Server

General Limitations

Table 16 lists recommended limitations for a Chat (Genesys Web Media) solution running on a single host with two Intel Xeon 3.0GHz processors.

Table 16: Chat Server Limitations

Item	Maximum	Rare Temporary Peaks
Message size	2 KB	4 KB (Genesys Desktop limitation)
Transcript size	54 KB	54 KB (Genesys Desktop limitation)
Concurrent sessions (in a realistic simple scenario)	500 per Chat Server	Not applicable
Messages per second	50	120
Sessions opened and closed per second	5	25
Languages in a session	Not applicable	No limit

In Table 16, *optimum* means without significant delay.

Chat Server also has a timeout that you can configure using the `user-register-timeout` option (default value 30 seconds). This is the maximum time between opening a socket and either of the following:

- Receiving a registration over the socket
- Receiving a flex packet over the socket

This timeout prevents keeping unused connections open.

Matching Contact Attributes

When a home user asks to open a chat session, the web interface gets him or her to fill in some identifying information, such as e-mail address, phone number, first name, last name, an so on.

This identifying information becomes a part of the *user data* that is associated with the interaction. The web interface relays this user data to Chat Server, and Chat Server sends it to UCS.

UCS then looks to see if the home user matches any of the people that it has represented as contacts in its database. It does this according to the following algorithm:

Table 17: UCS Search Order for Contact Records

Attribute Name	Search Order
EmailAddress	0
PhoneNumber	1
FirstName	2
LastName	2

UCS is hard-coded to use this algorithm with interactions coming from Genesys media servers, namely e-mail, chat, and callback interactions. For other media the algorithm can be customized.

So if the user data includes an attribute called `EmailAddress`, UCS looks for a contact in its database whose `EmailAddress` attribute has the same value as the user data attribute. (For details on the structure of this part of the UCS database, see the "Contact Package" chapter in *Multimedia 7.6 Selected Conceptual Data Models for the UCS Database.*) The name of the user data attribute must be exactly `EmailAddress`—if it is `email_address` or anything else, UCS will not try to match its value with the stored value of `EmailAddress`.

If UCS finds no matching contact, it creates a new one using the user data (see Chapter 6, "Contact Identification and Creation," on page 197 for more information).

For either a matching contact or a new one, UCS sends the following, as data about the contact for this interaction, to Chat Server:

- The matched attribute (if not e-mail address, then phone number, and so on).
- The attribute `ContactID`.
- All other attributes of this contact that UCS has stored in its database, except:
- If any user data has an attribute name that matches an attribute name in the UCS `Contacts` table, UCS returns the value of the attribute from the user data, not the value from the `Contacts` table. It does not modify the value in the `Contacts` table.

The last point can cause a problem, as in the following example:

1. Home user Steve Jones wants to open a chat session. In the web interface, he types in his correct e-mail address sjones@here, then erroneously types his first name as Speve.

2. UCS finds a contact record for sjones@here.
3. UCS returns to Chat Server data about an existing contact whose e-mail address is sjones@here and whose first name is Speve. UCS still has the correct first name Steve in its database, but the user data, with the erroneous Speve, preempts the correct data for the purposes of this chat interaction.
4. The system uses the user data to generate the message prompt that marks the home user in the chat display. As a result, the chat session displays something like the following:

   ```
   14:52:20 SpeveJ has joined the session
   14:52:30 SpeveJ > Hi.
   ```

5. The Agent Desktop displays the incorrect first name (in the user data on the lower left pane) and the correct first name (on the Customer Records pane on the right). The agent sees the incorrect first name and opens the chat session by typing, “Hello Speve, how can I help you?”
6. The interaction passes through a strategy that generates an automatic response, which opens, “It was good chatting with you, Speve.”

To avoid this type of problem, be sure that the system (including strategies and desktop) as well as its users refer to the UCS database, rather than user data, for contact attributes. In the example just cited, the agent must be sure to look at the Customer Records (right-hand) pane of the Desktop for the name of the contact. However, it is not possible to avoid the use by the system of user data to generate the message prompt (`SpeveJ` in the example).

It is also advisable to closely monitor the inventory of contact attributes that can become user data.

Attached Data

This refers to data that is attached to the interaction during processing, for example by the media server that creates the interaction, or by a routing strategy. Interaction Server places no limit on such data, but its size does have an effect on performance. Genesys recommends that you limit attached data to a maximum of 16 KB per interaction.

In general, attach only data that is needed for routing and/or reporting; do not attach data if you are not sure it will be used.

Interaction Server

You should be aware of the following:

- Interaction Server does not support the following requests:
  - RequestQueryServer
  - RequestQueryLocation

  - RequestDeletePair (when URS sends this request after RequestRouteCall)
- It is not desirable to run Interaction Server in an environment in which servers and clients differ as to the codepages used (by operating systems or databases). In such an environment, characters of non-Latin alphabets may appear as the symbol ? (question mark) in log files and in applications with a user interface, such as Agent Desktop. The functionality of other features of the solution may also be restricted or compromised.

E-mail Server Java

There is no limit on the size of attachments to e-mails. You can use the `maximum-msg-size` option to limit the overall size of incoming messages (that is, the total size of all message parts, including the body and any attachments).

Co-Browsing Server

Co-browsing clients support the HTTPS protocol only.

Unicode Character Support

Although UCS supports Unicode character sets, other components of Multimedia and of the Genesys suite (in particular, Interaction Server and URS) do not. This means that interactions that use a Unicode character set may be corrupted. Specifically, what may be corrupted is any part of the interaction's data that is handled by Interaction Server or URS (or any other component that does not support Unicode). This includes attributes such as Subject, FirstName, and LastName. It does not include the body of the interaction, which is handled by UCS only.

The following scenario provides an example of how this corruption can happen:

1. URS processes an interaction that includes Unicode user data, such as Subject. Because URS does not support Unicode, the Subject and other user data is corrupted.
2. UCS receives `RequestStopProcessing`, either from URS or the agent desktop.
3. UCS saves the interaction's user data (this is done in case the user data has changed during processing), copying certain properties from the user data into the corresponding fields of its `Interaction` table.

   This user data includes the interaction's subject, the value of which is copied into the `Subject` attribute. But the user data was corrupted during processing by URS, so the corrupted data is now stored in the UCS database.

4. If the corrupted subject data is used to compose another e-mail (such as reply or redirect), the subject of the new e-mail is also corrupted.

As a workaround for this scenario, you can modify the applicable strategy so that it deletes the Subject user data before it issues `RequestStopProcessing`.

E-mail Server Java: Advanced Topics

This section provides information on various topics relevant to E-mail Server Java.

Handling Unparsable E-Mails

If E-mail Server Java is unable to parse an incoming e-mail, it creates a new e-mail interaction (a "wrapping message") with the following characteristics:

- The header is the same as the header of the original, unparsable e-mail.
  - If the header of the original e-mail is unparsable, the subject of the new interaction is `Unknown subject`.
  - If the From address of the original e-mail is not valid, the From address of the new interaction is `unknown@<default_domain>`, where <default_domain> is the domain specified by the `default_domain` configuration option of the E-mail Server Java application.
- The text of the new interaction is `Error encountered during preprocessing of this message + <reason_for_failure> + Original Incoming Email is attached to this Email.`
- The original e-mail is attached to the new e-mail.
- The new e-mail has an attached key-value pair, whose key is `_WrappingMessageReason` and whose value is a text string that describes the reason for creating the wrapping message.

JavaMail Properties

E-mail Server Java uses the JavaMail API library 1.4. JavaMail can make use of numerous properties, which are documented at the following locations:

- Environment properties: http://java.sun.com/products/javamail/JavaMail-1.4.pdf (Appendix A: Environment Properties)
- JavaMail Session properties: http://java.sun.com/products/javamail/javadocs/overview-summary.html
- JavaMail Session properties for IMAP: http://java.sun.com/products/javamail/javadocs/com/sun/mail/imap/package-summary.html

- JavaMail Session properties for POP3: http://java.sun.com/products/javamail/javadocs/com/sun/mail/pop3/package-summary.html

These properties are treated in different ways in Multimedia, depending on

- Whether they are set internally by E-mail Server Java.
- Whether they can be modified by users.

These two parameters define three different categories of property:

- Set internally and not user-modifiable
- Set internally and user-modifiable
- Not set internally and user-modifiable

The next three sections list the properties in each category and describe how to set the ones in user-modifiable categories.

Set Internally, Not User-Modifiable

mail.pop3.class

mail.imap.class

Set Internally, User-Modifiable

mail.debug

mail.pop3.host

mail.pop3.user

mail.pop3.port

mail.pop3.connectiontimeout

mail.pop3.timeout

mail.pop3.socketFactory.class

mail.pop3.socketFactory.fallback

mail.pop3.socketFactory.port

mail.imap.host

mail.imap.user

mail.imap.port

mail.imap.connectiontimeout

mail.imap.timeout

mail.imap.socketFactory.class

mail.imap.socketFactory.fallback

mail.imap.socketFactory.port

You can modify these using existing configuration options, as shown in the following table. In this table, <protocol> is either POP3 or IMAP; for example, `mail.<protocol>.timeout` covers `mail.pop3.timeout` and `mail.imap.timeout`.

Table 18: JavaMail Properties Controlled by Configuration Options

JavaMail Property	Configuration Option
mail.debug	enable-debug
mail.<protocol>.connectiontimeout	connect-timeout
mail.<protocol>.timeout	protocol-timeout
mail.<protocol>.user	mailbox
mail.<protocol>.host	server
mail.<protocol>.port mail.<protocol>.socketFactory.port	port
mail.<protocol>.socketFactory.class mail.<protocol>.socketFactory.fallback	enable-ssl

See the *Multimedia 7.6 Reference Manual* for complete information on these options.

Not Set Internally, User-Modifiable

Any of the properties not listed in the two preceding sections can be modified by creating options in E-mail Server Java's `pop-client` section. The option name is the property name. For the value, see the JavaMail documentation listed above.

Note: Do *not* use this method to modify the properties, listed in the preceding section, that are controlled by configuration options.

Here is an example of adding an option to modify a JavaMail property: Some POP3 servers do not properly implement TOP, an optional POP command. This can create conflicts between the results of the TOP and RETR commands, which in turn can prevent E-mail Server Java from parsing the retrieved e-mail. To prevent these conflicts, you can create an option that invokes JavaMail's mail.pop3.disabletop property. The option name is `mail.pop3.disabletop`, it must be in the `pop-client` section, and its value must be `true`. E-mail Server Java then does not use TOP to retrieve messages, only RETR.

Delivery Status Notification and Message Disposition Notification

Outbound e-mails can include a request for a return message indicating whether and how the original e-mail was delivered. In Genesys Multimedia, you do this using the `Send Email` object in a routing strategy, as described in the *Universal Routing 7.6 Reference Manual.* The return message is of one of the following three types:

- If delivery fails: InboundNDR
- If delivery succeeds:
  - InboundReport
  - InboundDisposition

These types are represented as attribute values of the `Interaction Subtype` Business Attribute in Configuration Manager. E-mail Server Java assigns the return message to the appropriate type, and UCS stores it as a child of the outbound e-mail that contained the request.

The following sections describe each type and its contents.

InboundNDR

If one of the SMTP servers involved in the transport of the original e-mail fails to deliver it, E-mail Server Java submits the return message to the system with subtype `InboundNDR` (NDR stands for non-delivery report). There are two ways that E-mail Server Java can detect an inbound e-mail as an NDR:

- The e-mail's From field contains one of the values specified in the E-mail Server Java `ndr-senders-list` option (the default is `mailer-daemon,postmaster,mmdf`). In this case, E-mail Server Java submits the e-mail to the system with no particular attached data. The subtype `InboundNDR` indicates the failure of delivery; the message itself contains no additional information.
- (1) The e-mail conforms to RFC 1894, which means that it includes information about delivery status, and (2) that information indicates that delivery failed for at least one recipient. In this case, E-mail Server Java submits the e-mail to the system with attached data.

Structured Information

When E-mail Server Java attaches data to the inbound interaction, it is of two kinds. The first kind, structured information, is listed in Table 19.

Table 19: Attached Data: Structured Information

Key	Possible Values	Description
_DSNInfo.RecipientCount	Any integer	Number of recipient addresses covered in this DSN
_DSNInfo.Recipient1.Recipient	Any string	Recipient address
_DSNInfo.Recipient1.Action	delayed delivered expanded failed relayed	Action applied for this recipient
_DSNInfo.Recipient2.Recipient	Same as _DSNInfo.Recipient1.Recipient	
_DSNInfo.Recipient2.Action	Same as _DSNInfo.Recipient1.Action	
_DSNInfo.RecipientN.Recipient	Same as _DSNInfo.Recipient1.Recipient	
_DSNInfo.RecipientN.Action	Same as _DSNInfo.Recipient1.Action	

In RecipientN in the last two rows of Table 19, N is the value of `_DSNInfo.RecipientCount`: the number of recipients covered in this InboundReport.

Note: A non-delivery report may arrive even if the outbound e-mail did not request it. If it does, E-mail Server Java still submits it to the system with the subtype `InboundNDR`.

Raw Information

The second kind of information that E-mail Server Java attaches is raw information. That is, all information included in the reply e-mail's header is attached as key-value pairs, with the key name formed by prefixing `_DSNRawInfo` to the field name used in the reply. Some examples are:

```
_DSNRawInfo.Reporting-MTA
_DSNRawInfo.RecipientCount
_DSNRawInfo.Recipient1.Original-Recipient
_DSNRawInfo.Recipient1.Action
_DSNRawInfo.Recipient1.Status
_DSNRawInfo.Recipient1.Remote-MTA
_DSNRawInfo.Recipient2.Final-Recipient
```

For details, see RFC 1894 (`http://www.ietf.org/rfc/rfc1894.txt`).

InboundReport

You request delivery status notification (DSN) by selecting the `Delivery status notification` box in a `Send Email` routing object. The reply to this request receives the subtype `InboundReport`. This reply conforms with RFC 1894, and includes, as attached data:

- The structured information listed in Table 19 on page 186.

  Note that if the `_DSNInfo.RecipientN.Action` key has a value of `failed`, E-mail Server Java assigns the reply the subtype `InboundNDR`, not `InboundReport`.
- The raw information contained in the keys whose names start with `_DSNRawInfo`, as described in the previous section.

Note: Depending on the implementation of the SMTP servers involved, there are the following possibilities:

- Individual DSN messages can be generated, each one related to one or several recipient addresses.
- A single DSN message can be generated, related to one or several recipient addresses.

InboundDisposition

You request message disposition notification (MDN, also called *read receipt*) by selecting the `Message disposition notification` box in a `Send Email` routing object. The reply to this request receives the subtype `InboundDisposition`. This reply conforms with RFC 3798, and includes, as attached data, the information listed in Table 20.

Table 20: Attached Data in InboundDisposition

Key	Possible Values	Description
_MDNInfo.ActionMode	manual-action automatic-action	Mode of the action applied to the e-mail
_MDNInfo.DispositionType	displayed deleted	What was done with the e-mail

Table 20: Attached Data in InboundDisposition (Continued)

Key	Possible Values	Description
_MDNInfo.SendingMode	MDN-sent-manually MDN-sent-automatically	How the message disposition notification is being sent
_MDNInfo.Recipient	Any string	Recipient address covered by this message disposition notification

This reply is sent as long as all of the following conditions are met:

- It was requested in the outbound e-mail. This is independent of whether a delivery status notification was also requested.
- Delivery succeeded.
- Either of the following:
  - The recipient agreed to send the read receipt.
  - The recipient mailer was configured to automatically send read receipts.

In addition to the structured information listed in Table 20, all information included in the reply e-mail's header is attached as key-value pairs, with the key name formed by prefixing `_MDNRawInfo` to the field name used in the reply. Some examples are:

```
_MDNRawInfo.Disposition
_MDNRawInfo.Final-Recipient
_MDNRawInfo.Original-Message-ID
_MDNRawInfo.Reporting-UA
```

For details, see RFC 3798 (`http://www.ietf.org/rfc/rfc3798.txt`).

Interaction Server: Advanced Topics

Custom Database Fields

The ability to create custom interaction properties has been expanded in release 7.6.1. You can now create new interaction properties (fields) without being limited to using the `CustomString<n>` and `CustomNumber<n>` fields that existed in the `interactions` table in release 7.6.0 and earlier. Also, the data type of these custom properties can be timestamp, in addition to the string and number types previously available.

Procedure: Configuring a custom interaction property

Start of procedure

1. Decide on an attached data key that will be the source of the content of the custom property.
2. Create a new field directly in the `interactions` database.
3. Create a corresponding Configuration Server database object:
   a. Create an `InteractionCustomProperties` Business Attribute if it does not already exist (as described in Step 2 on page 212).
   b. Give it an `Attribute Value`, with a name exactly matching the attached data key name that you decided on in Step 1. The matching is case sensitive (you can create a separate display name).
   c. In your new attribute value, go to the `Annex` tab and create a section called `translation`.
   d. In the new `translation` section, create an option called `translate-to`, with its value duplicating the name of the new field you created in Step 2.

End of procedure

Note: If you specify a custom field as `not null`, you must ensure that applications always provide some data to that field upon creation of an interaction (RequestSubmit). If no data is provided, the request will fail because Interaction Server sends NULL for empty fields, and that will be rejected by the DBMS. This also means that the `default value` trigger for such fields cannot be used.

As a workaround, you can create fields in the `interactions` database without mapping them to custom properties of Interaction Server (as described in Steps 1and 3). Such fields are hidden from Interaction Server but can be used by third-party applications.

Converting from BLOB

Interaction Server ordinarily stores attached data in the `flexible_properties` field as a BLOB (binary large object). You can convert attached data to a custom field by running Interaction Server in a special utility mode, in which Interaction Server uses the key-value format of this attached data to convert all such fields to custom fields.

To run Interaction Server in utility mode, launch it from a command line with the following option:

```
-convert-fields [command_or_parameters]
```

where the optional `command_or_parameters` is one of the following:

`reset`—Ensures that the next run in utility mode will start processing from the beginning, rather than picking up where it left off.

`bulk-size=N`—Determines the number of records that are processed before committing the transaction. The default value is 100, valid values are any integer in the range 1–1000.

Here is an example command line:

```
interaction_server -host genesys_host -port 9876 -app IxnSrv05
-convert-fields reset
```

You can also have Interaction Server convert an existing database field into a BLOB, stored in the `flexible_properties` field. To do so, use the following procedure.

Procedure: Converting a field to a BLOB

Start of procedure

1. Open the corresponding Business Attribute Value in Configuration Manager.
2. In the `translate-to` section, add an option called `to-delete` and give it the value `yes`.
3. Run Interaction Server in utility mode, as described previously (page 190).

   Interaction Server, in utility mode, moves the content of all such fields into the `flexible_properties` field and leaves the custom field with an empty value.

   Note: When Interaction Server runs in utility mode all of its other features are disabled: it cannot process interactions or open ports for clients.

End of procedure

Event Logger

In release 7.6.1 Interaction Server includes Event Logger, a mechanism for storing reporting event messages in a database. You can configure it to store all reporting events or a selected subset. You can also create multiple instances of it.

Interaction Server generates, to registered reporting engines, messages that provide a detailed picture of the processing of each interaction. The attributes of these messages include much information about the interaction itself, such as its type, time received, associated agents, queues and workbins it was placed it, and so on. For a reference listing of these events and their attributes, see the "Reporting Messages" section of the "Other Protocol Events Used by Interaction Server" chapter of the *Genesys 7 Events and Models Reference Manual.*

All configuration for the logger functionality is done in the Database Access Point (DAP) associated with the logger database.

Procedure: Deploying Event Logger

Start of procedure

1. Create a database to store the reporting data.
2. Locate the correct setup script for your RDMBS and run it on the database you created in Step 1.

   This script is called `eldb_<database_name>.sql`, where <database_name> is either *db2, mssql,* or *oracle* (for example, `eldb_mssql.sql`). To locate the script, go to the `Script` subdirectory of the installation directory of your Interaction Server, then open the subdirectory named after your RDBMS; for example, `\InteractionServer_761\Script\Oracle`.
3. Create a Database Access Point (DAP), filling in the usual mandatory settings on the `General` and `DB Info` tabs.
4. On the DAP's `Options` tab, create a section called `logger-settings`. This is the only mandatory section; its existence tells Interaction Server to use this DAP for storing reporting events.
5. In the `logger-settings` section, add at least one of the options described in "Options List" on page 193 (the section must contain at least one option in order to be valid).
6. Optionally add one or both of the following section types:
   - `event-filtering`—Contains options filtering out certain classes of event messages.
   - Custom data sections—Three sections that enable you to map the key name of any user data key-value pair onto a custom field in the Logger database.

   See "Options List" on page 193 more information.
7. On Interaction Server's `Connections` tab, add a connection to the DAP.

For multiple instances of the Event Logger, run the creation script multiple times and create multiple DAPs.

End of procedure

Notes: Genesys supplies a set of scripts that deletes events from the `rpt_interaction` table as soon as processing of the interaction stops. If you want to preserve this data, you can disable these scripts. Or you can disable the trigger `trg_delete_log_stopped_ixns` after you run the setup script. You can reenable it any time and resume removing records from the database automatically.

Of course event messages increase rapidly in number as interactions are processed, so you will want to take measures to periodically delete data from the database or move it elsewhere.

Also note that after creating or removing custom fields in a database, some triggers become invalid. If this happens, you must recompile them to be sure they work properly.

Classification of Events

The logger functionality classifies reporting events in two ways:

- By activity type—that is, whether the activity refers to an interaction, an agent, or an ESP server (see "ESP" on page 243). The database contains one table for each type: interaction activity is stored in `rpt_interaction`, agent activity is stored in `rpt_agent`, and ESP server activity is stored in `rpt_esp`.
- By endpoint type—that is, whether that interaction is being transmitted to a queue, strategy, agent, or ESP service. You can filter out events according to endpoint type, as described in "event-filtering Section" on page 194. A few events do not have an endpoint type; you cannot filter these events.

Table 21 lists the events and their classifications.

Table 21: Classification of Reporting Events

Event	Activity	Endpoint
EventPropertiesChanged	Interaction	-
EventPartyAdded	Interaction	Agent, Strategy
EventPartyRemoved	Interaction	Agent, Strategy
EventRevoked	Interaction	Agent
EventInteractionSubmited	Interaction	-

Table 21: Classification of Reporting Events (Continued)

Event	Activity	Endpoint
EventProcessingStopped	Interaction	-
EventPlacedInQueue	Interaction	Queue
EventPlacedInWorkbin	Interaction	Queue
EventAgentInvited	Interaction	Agent
EventRejected	Interaction	Agent
EventTakenFromQueue	Interaction	Queue
EventTakenFromWorkbin	Interaction	Queue
EventAgentLogin	Agent	Agent State
EventAgentLogout	Agent	Agent State
EventDoNotDisturbOn	Agent	Agent State
EventDoNotDisturbOff	Agent	Agent State
EventMediaAdded	Agent	Agent State
EventMediaRemoved	Agent	Agent State
EventNotReadyForMedia	Agent	Agent State
EventReadyForMedia	Agent	Agent State
EventAgentStateReasonChanged	Agent	Agent State
EventMediaStateReasonChanged	Agent	Agent State
EventExternalServiceRequested	ESP Server	ESP Server
EventExternalServiceResponded	ESP Server	ESP Server

Options List

This section provides short descriptions of the DAP options that configure the Event Logger's behavior. See the *Multimedia 7.6 Reference Manual* for full details.

logger-settings Section

`batch-size`—Defines the minimum number of records to store in internal memory before flushing to the database. Valid values are 1–5,000; the default is 500.

`max-queue-size`—Defines the maximum number of records that are kept in memory while waiting to be written to the database. If the number of records exceeds this maximum, the data are discarded from memory and are not written to the database. Valid values are 10,000–100,000; the default is 20,000.

`storing-timeout`—Defines a time interval, in milliseconds, between operations of writing to the database. Valid values are 500–60,000; the default is 1,000.

Note: `storing-timeout` and `batch-size` define limits that trigger writing to the database: writing takes place as soon as one or the other is reached.

`mandatory-logging`—Reserved for future use.

event-filtering Section

With the exception of `log-userdata`, each option in this section is named for one of the endpoint types listed in Table 21 on page 192. With the value `false`,

- For the options named for an endpoint type, events associated with the named endpoint type are filtered out.
- For `log-userdata`, data from custom fields is filtered out.

`log-agent-state`

`log-agent-activity`

`log-queue`

`log-strategy`

`log-userdata`

`log-esp-service`

For example, setting `log-queue` to a value of `false` prevents the events `EventPlacedInQueue`, `EventPlacedInWorkbin`, `EventTakenFromQueue`, and `EventTakenFromWorkbin` from being stored.

itx-custom-data, esp-custom-data, esp-service-data Sections

These three sections contain options specifying a list of user data keys that are to be stored in custom fields of the event logger database. All three sections work identically, the differences being (a) the events from which the user data

is taken and (b) the database table that stores them. These differences are shown in Table 22.

Table 22: Custom Data Sections

Section	Source Event	Logger Database Table
`itx-custom-data`	All reporting events other than `EventExternalServiceRequested` and `EventExternalServiceResponded`	`rpt_interaction`
`esp-custom-data`	`UserData` attribute of `EventExternalServiceRequested` and `EventExternalServiceResponded`	`rpt_esp`
`esp-service-data`	`Envelope3rdServer` attribute of `EventExternalServiceRequested` and `EventExternalServiceResponded`	`rpt_esp`

For an explanation of the `Envelope3rdServer` attribute, see "EventExternalServiceRequested Attributes" and "Structure of Envelope3dServer Attribute" in the "Reporting Messages" section of the "Other Protocol Events Used by Interaction Server" chapter of the *Genesys 7 Events and Models Reference Manual.*

To use these options, you must first add a field to the appropriate Event Logger database table. Its data type must be the same as that of the mapped user data key.

In these sections, the options have the following characteristics:

- The name is a user data key name (case-sensitive).
- The value is three semicolon-separated strings, which specify the following:
  - **a.** The name of the field that you added to the database table. This value is required.
  - **b.** The data type: string, integer, or timestamp. The default is string, with default length 64. If your data type is other than string, or if it is string and you want to specify a non-default length (next item), this value is required.
  - **c.** Optionally, the length. The default for the string type is 64. There are no default values for integer and timestamp.

For example, if you have a data key called CustomerSegment, you can add a custom field to store this data as follows:

1. Add a field called `customer_segment` to the `rpt_interact` table.
2. In the `itx-custom-data` section, create an option called `CustomerSegment`.
3. Give it this value: `customer_segment;string;64`.

Since string and 64 are the default values for type and length respectively, the value of this option could also be simply `customer_segment`.

Chapter

6 Contact Identification and Creation

This chapter describes how UCS identifies and creates new contacts, in these sections:

- Overview, page 197
- Contact Identification, page 197
- Contact Creation, page 203

Overview

When a new interaction enters the system, UCS performs the following tasks:

1. Contact identification—UCS checks whether this interaction is coming from a known contact: more precisely, whether the contact data included in the new interaction matches an existing contact in its database. It does this in response to a request from a media server, the `Identify Contact` Routing strategy object, or the Open Media Interaction `CreateNew` block.
2. Contact creation—If the contact does not exist in the database, UCS creates a new record to represent it.

This chapter describes the default procedure that UCS follows for these tasks, and ways in which you can modify this default.

Contact Identification

To perform contact identification, UCS takes the contact data included in the new interaction and tries to match it with its existing contact records based on certain attributes.

Default Behavior

What UCS Checks

The default list of the attributes that UCS checks is `FirstName`, `LastName`, `Title`, `EmailAddress`, and `PhoneNumber`.

Note: This chapter refers to contact attributes by their system names. The system name is the one that is used in the UCS and Configuration Server databases, and the one that appears in the `Name` box of the attribute's `Properties` window in Configuration Manager. In Configuration Manager there is also a display name which usually differs slightly; for example, system name `PhoneNumber`, display name `Phone Number`.

Order of Checking

The attributes that UCS uses in contact identification have a ranking which tells UCS what priority to give them in searching. The default ranking is:

`EmailAddress`—0

`PhoneNumber`—1

`FirstName`—2

`LastName`—2

`Title`—3

This means the following (using *target contact* to mean the contact associated with the new interaction, the one that UCS is trying to identify in its database):

- If UCS finds just one existing contact whose `EmailAddress` attribute matches that of the target contact, it stops and associates the new interaction with the existing contact.
- If UCS finds no existing contact with a matching `EmailAddress`, it stops searching and proceeds to create a new contact record.
- If UCS finds two or more existing contacts with matching `EmailAddress` attributes, it adds in the attribute with the next-highest priority and tries to match the `PhoneNumber` attribute as well as the `EmailAddress`.
- With the combination of the `EmailAddress` and `Phone Number` attributes, UCS again pursues the same three alternatives:
  - If exactly one contact (among those already retrieved as matching the preceding attribute) matches these two attributes, the existing contact is associated with the new interaction.
  - If no contacts match, UCS creates a new contact.
  - If two or more contacts match, UCS adds in the attribute with the next-highest priority.
- When two attributes have the same rank, as do `FirstName` and `LastName` in the default ranking, UCS tries to match both of them.

  If the interaction has values for both attributes, UCS returns only contact records that match both. But if one of equally-ranked attributes has no value, UCS returns any contact records that match the attribute that does

have a value (this is equivalent to saying that an attribute with no value matches everything).

If UCS goes through all of the attributes that it has been specified to check and still finds two or more matching contacts, the next step depends on the entity that requested the contact identification:

- If UCS is identifying contacts in response to a request from a media server:
  - With E-mail Server Java, UCS takes the first contact on the list and associates it with the interaction.
  - With Chat Server, UCS simply passes the interaction on for processing with no associated contact. It reports neither the list of contacts found nor the fact that multiple contacts were found. The agent handling the interaction can select a contact for it using the Agent Desktop.
- If UCS is identifying contacts in response to an IRD `Identify Contact` object, it depends on whether the object's `Return Unique` checkbox is selected:
  - If `Return Unique` is not selected, UCS returns a list of the matching contacts.
  - If `Return Unique` is selected, UCS only reports that multiple contacts were found but does not return the list of matching contacts.

  In either case, what happens next depends on the subsequent part of the strategy. The system may continue to process the interaction without any contact, until the interaction reaches the Agent Desktop, when the agent handling the interaction can select a contact for it.

Customizing

You can alter UCS's default contact identification behavior in both aspects (whether an attribute is used and, if it is used, what priority it has in matching). You can do this for all interactions globally or for interactions of a specific media type.

With these differing scopes of customization, the more specific scope takes precedence. UCS operates according to the following order:

1. If a media type has its own configuration for how a contact attribute is used in contact identification, that takes precedence over any other configuration ("Customize per media type" on page 201).
2. Configuration of a contact attribute itself is more general and applies only to those media types that lack their own configuration for contact identification ("Customize which attributes are checked" on page 200 and "Customize the priority of attributes" on page 200).
3. In any areas where items 1 and 2 do not apply, UCS follows the default behavior already described (page 198).

Procedure: Customize which attributes are checked

Start of procedure

1. In Configuration Manager, open the `Properties` window for the desired attribute.
2. On the `Annex` tab of the `Properties` window (to set Configuration Manager to show the `Annex` tab, see `Configuration Manager Help`), create a `settings` section if it does not already exist.
3. In the `settings` section, create an option called `is-searchable`. Set its value to `TRUE` to make UCS use this attribute in contact identification. Set its value to `FALSE` to keep UCS from using this attribute in contact identification.

   (The default value depends on the attribute. The five attributes used in UCS's default behavior (page 198) have the default value `TRUE`. All other attributes have the default value `FALSE`).

Note: This procedure affects the value of the desired attribute's `IsSearchable` attribute in the UCS database. You must use only this procedure to do this. Never edit any attribute or value directly in the database.

End of procedure

Procedure: Customize the priority of attributes

Summary

You can customize the priority of the attributes that UCS checks, but only when both of the following conditions are true:

- The media type is specified in the interaction's user data.
- The interaction's media type is *not* e-mail, chat, or callback. In the typical case, this would be because the interactions are being submitted by a custom media server (built using the Genesys Media Interaction SDK Java or Open Media Interaction SDK Web Services).

Start of procedure

1. In Configuration Manager, open the `Properties` window for the desired attribute and go to its `Annex` tab. Create a `settings` section if it does not already exist.
2. In the `settings` section, create an option called `search-order-level`. Give the option a numerical value to determine its priority in matching. Possible values are any integer in the range 0 (highest priority) to 127 (lowest priority). Two or more attributes can have the same priority.

End of procedure

Procedure: Customize per media type

Summary

For a finer level of granularity, you can control how contacts are identified for interactions of a specific media type (of course, the interactions must have a valid media type specified in their user data).

Start of procedure

1. In Configuration Manager, open the `Properties` window for the desired media type and go to its `Annex` tab.
2. Create a `contact-searchable-attributes` section.
3. In this section, create an option for each contact attribute that you want UCS to use in contact identification for this media type.

   Note: Any such attribute must either be one on the default list (page 198) or have the `is-searchable` option with a value of `TRUE`.

   The options that you create must have the following characteristics:
   - The name duplicates the attribute's system name (displayed in the `Name` box of the `Properties` window, not the `Display Name` box); for example, `PhoneNumber` rather than `Phone Number`.
   - The value has the form "`level=<integer>;mandatory=<Boolean>`", where
     - `<integer>` is an integer from 0–127 setting the attribute's priority.
     - If `<Boolean>` is `TRUE`, this attribute must be present (that is, it must have a value) for matching to occur.
     - If `<Boolean>` is `FALSE`, this attribute need not be present (that is, it may lack a value) for matching to occur.

Figure 84 shows an example.

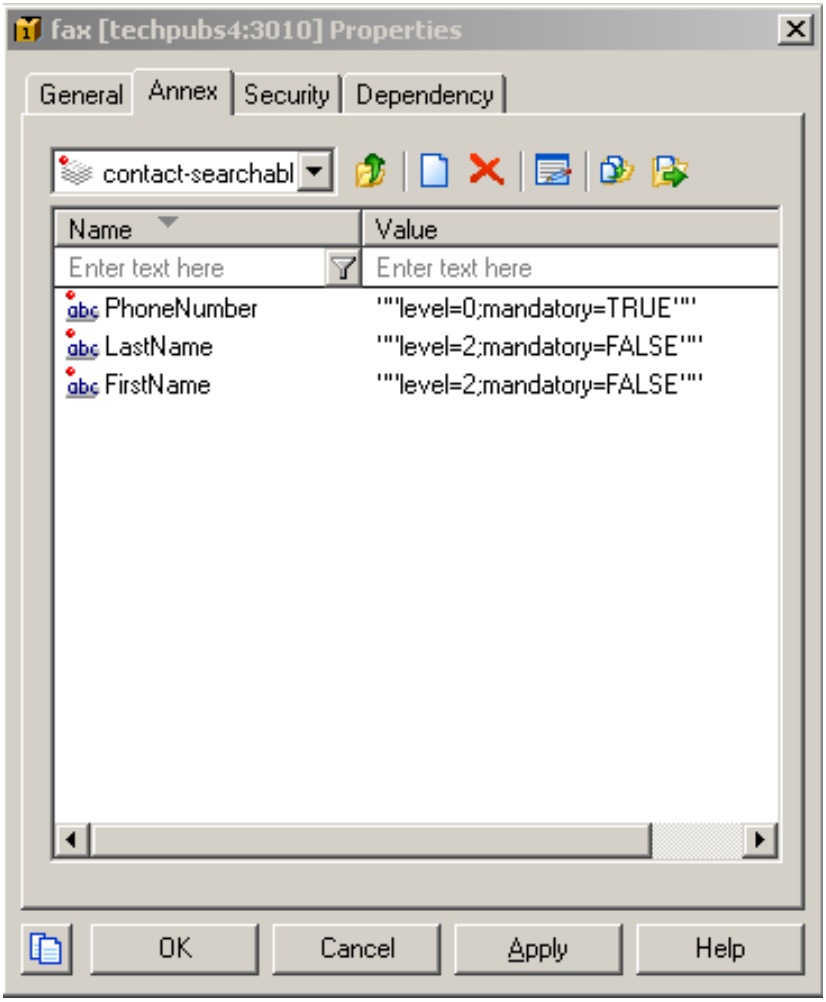


Figure 84: Media Type Configured for Contact Identification

With the configuration in Figure 84,

- UCS returns an error if no `PhoneNumber` value is present. It does this even if `FirstName` and `LastName` values are available, because `PhoneNumber` is the attribute with highest priority (0) and is defined as mandatory.
- If multiple matching records are found with a given `PhoneNumber` value, UCS uses `FirstName` and `LastName` together to discriminate because they have the same priority (2).
- If there is no `LastName` value, UCS does not use `FirstName` alone, because `LastName` is specified as mandatory in this level. Instead, the list of records matching `PhoneNumber` only is returned. This avoids identification based on `FirstName` only.

End of procedure

Contact Creation

If UCS cannot find a contact in its database that matches the contact associated with a new interaction, it creates one in its database.

Default Behavior

The default behavior is for UCS to simply create a new contact record. This behavior can be overridden for an interaction in a routing strategy using the IRD `Identify Contact` object (see *Universal Routing 7.6 Interaction Routing Designer Help*), or by using one of the non-default settings of E-mail Server Java's `contact-identification` option (see the *Multimedia 7.6 Reference Manual*).

Customizing

Procedure: Turning contact creation on and off

Summary

You can customize the default behavior for each media type.

Start of procedure

1. In Configuration Manager, go to `Business Attributes > Media Type > Attribute Values` and double click the media type that you want to adjust.
2. On the `Annex` tab of the resulting `Properties` window (to set Configuration Manager to show the `Annex` tab, see `Configuration Manager Help`), create a `settings` section if it does not already exist.
3. In the `settings` section, create an option called `create-contact`. Set its value to `FALSE` to block contact creation for this media type. (The default value is `TRUE`).

End of procedure

Procedure: Setting minimum attributes for creation

Summary

You can also define a minimum set of contact attributes that must be present (must have values) for a contact to be created.

Start of procedure

1. In Configuration Manager, open the `Properties` window for the desired media type and go to its `Annex` tab.
2. Create a `contact-minimum-attributes-set` section.
3. In this section, create an option for each contact attribute that you want to require for contact creation. The option's name must duplicate the attribute's database name (displayed in the `Name` box of the `Properties` window, not the `Display Name` box); for example, `PhoneNumber` rather than `Phone Number`. The option's value must be empty.

End of procedure

Chapter

7

Interaction Properties

This chapter describes interaction properties and provides information on which of them are safe to modify. It covers these topics:

- Overview, page 205
- Interaction Properties, page 207
- Business Attributes, page 214
- Translations, page 216

Overview

Genesys Multimedia interactions have a number of properties that take the form of key-value pairs. This chapter lists these properties and provides some information about them, including whether it is safe to change them.

Since you can set up your Multimedia system to change some of these properties as the interaction moves through the system, it is particularly important to know that some properties are safe to change and others are not. For example, you can change interaction properties by using the `Update`, `UpdateBusinessData`, and `UpdateInteractionData` functions in a routing strategy. But you should be very careful when considering using these functions to change any interaction property.

There are four types of properties that can be defined in terms of the following two characteristics

- Whether they exist as independent fields in the `interactions` table. If they do, you can refer to them when defining conditions, orders, and segments in Views in Business Processes.
- Whether they are used by Genesys media servers.

Table 23 summarizes this classification and specifies whether it is safe for users to change the properties.

Note: When this chapter says that you must not change an interaction property, that means that you must not change the value of existing properties *and* that you must not create a new object with this name.

Table 23: Four Property Types

Type	Independent Field	Used by Genesys Media Server	Change OK
System	+	-	No
Business	+	+	Yes
Custom	+	-	Yes
Other	-	+	No

Note: Significant changes were made to the `interactions` table in release 7.6.1. "Changes in Release 7.6.1" on page 213 describes these changes. Other locations in this chapter provide cross-references to this section where relevant.

Here is further information on these four property types:

- System properties. These are maintained solely by Interaction Server, as independent fields in the `interactions` table. The user cannot change them. The definition of conditions, orders, and segments in Views can refer to them. Examples: `InteractionID, MediaType`. See also "System Properties" on page 207.
- Business properties. The values of these are set by media servers (including both Genesys and custom media servers), and these properties may be used by the media servers at later points in processing. Therefore you must not change the ones that are used by any media server in your solution. They are stored as independent fields in the `interactions` table (this differs in release 7.6.1; see "Business Properties Not Stored as Independent Fields" on page 213). The definition of conditions, orders, and segments in Views can refer to them. Examples: `ContactID, Subject`. See also "Business Properties" on page 210.
- Custom properties. These are reserved as independent fields in the `interactions` table with generic names such as `CustomString3`. They are available for you to define as interaction properties. Media servers do not use them. can refer to them. See also "Custom Properties" on page 211 and "Flexible Configuration of Custom Properties" on page 214.

- Other properties. These are used by media servers. They are not stored as independent fields in the `interactions` table and can not be used to define Views. Users must not change them. See also "Other Properties" on page 213.

Note: In setting conditions for Views and snapshots, besides directly using some interaction properties types, you can also use a set of functions called Translations. These provide database manipulation tools that are independent of the underlying database. See "Translations" on page 216.

Interaction properties show up in several places:

- Interaction Server logs show all of them.
- Some IRD objects (for example, `UpdateBusinessData`) use them as parameters and display them in drop-down lists.

Interaction Properties

This section lists the following about interaction properties:

- Property name as it appears in protocol messages
- Field name in the `interactions` table
- Data type
- Short description

System Properties

Table 24 lists the system properties. You must not change these properties. You can use these properties on the `Condition`, `Order`, and `Segmentation` tabs of Views in Business Processes, except that properties with Timestamp data type cannot be used on the `Segmentation` tab.

Table 24: System Properties

Name	Name in Interactions Table	Type	Description
DeliveredAt	delivered_at	Timestamp	Date and time that the interaction was first offered to the agent
ExternalId	external_id	String	External interaction identifier (examples: chat session ID, e-mail mime ID)
InQueues	destinations	String	Suggested destinations for the interaction (if provided by URS)

Table 24: System Properties (Continued)

Name	Name in Interactions Table	Type	Description
InteractionId	id	String	Record identifier
InteractionState	state	Integer	`0` = queued `2` = being processed by URS `3` = being handled by agent
InteractionSubtype	subtype	String	Defined as `Business Attribute` in Configuration Server
InteractionType	type	String	Defined as `Business Attribute` in Configuration Server
IsLocked	is_locked	Integer	`0` = unlocked `1` = locked
IsOnline	is_online	Integer	`0` = offline `1` = online This property applies to any media type (for e-mail, the value is always `0`).
MediaType	media_type	String	Defined as `Business Attribute` in Configuration Server
MovedToQueueAt	moved_to_queue_at	Timestamp	Date and time that the interaction was first moved to a queue. If the interaction leaves the queue, then returns to it, the value of this property remains the same; that is, it shows the earliest time that the interaction entered the queue, without regard for later entry to or exit from the queue.
OutQueues	destinations	String	Suggested destinations for reply
ParentId	parent_id	String	Identifier of the parent interaction in the UCS database
PlacedInQueueAt	placed_in_queue_at	Timestamp	Date and time that the interaction was placed in the current queue, regardless of whether the interaction was in this queue previously
PlaceInQueueSeq	place_in_queue_seq	Integer	Event sequence number, denoting the chronological order of the last `EventPlacedInQueue` (for the interaction)

Table 24: System Properties (Continued)

Name	Name in Interactions Table	Type	Description
Queue	queue	String	Name of the queue that the interaction is in
ReceivedAt	received_at	Timestamp	Date and time that the media server received the interaction. If not provided by media server, date and time of submission to Interaction Server (same as `SubmittedAt`).
ScheduledAt	scheduled_at	Timestamp	Date and time before which the interaction must not be processed. See "Setting the ScheduledAt Property" in the "Creating Business Process Objects" chapter of the *Universal Routing 7.6 Business Process User's Guide.*
SubmittedAt	submitted_at	Timestamp	Date and time that the interaction was submitted to Interaction Server
SubmittedBy	submitted_by	String	Application name of the media server that submitted the interaction
SubmitSeq	submit_seq	Integer	Event sequence number, denoting the chronological order of `EventInteractionSubmitted` (for the interaction)
TenantId	tenant_id	Integer	Tenant associated with the interaction
Workbin	workbin	String	Indicates that interaction is in a workbin
WorkbinAgentGroupId	agent_group_id	String	One of four properties specifying the ID and type of the workbin that the interaction is in. Only one of the four properties is present.
WorkbinAgentId	agent_id	String	See `WorkbinAgentGroupId`.
WorkbinPlaceGroupId	place_group_id	String	See `WorkbinAgentGroupId`.

Table 24: System Properties (Continued)

Name	Name in Interactions Table	Type	Description
WorkbinPlaceId	place_id	String	See `WorkbinAgentGroupId`.
AbandonedAt	abandoned_at	Timestamp	Date and time that the media server set the `IsOnline` attribute to `0`. Simplifies calculation of some statistics. If the interaction is still online, this attribute is not set.

Business Properties

Table 25 lists business properties. You can use these properties on the `Condition` and `Order` tabs of Views in Business Processes. It is safe to change these properties.

Note: For significant changes to these properties in release 7.6.1, see "Business Properties Not Stored as Independent Fields" on page 213.

Table 25: Business Properties

Name	Name In Interactions Table	Type	Description
CaseId	case_id	String	Case identifier. Use and meaning to be defined by user.
CategoryId	category	String	Category identifier obtained by routing strategy request for e-mail classification
ContactId	contact_id	String	Customer identifier in the UCS database. Provided by UCS.
CustomerSegment	customer_segment	String	Code for the customer's revenue potential; for example, `Gold`, `Silver`, `Bronze`. May be assigned as a result of database lookup based on sender name. Routing strategy sets this property.
DispositionCode	disposition_code	String	Code for moving the interaction somewhere else
FromAddress	from_address	String	Taken from e-mail interaction

Table 25: Business Properties (Continued)

Name	Name In Interactions Table	Type	Description
FromPersonal	from_personal	String	Name of person who sent the interaction
Mailbox	mailbox	String	Mailbox of addressee
Priority	priority	Integer	Indicates whether e-mail should receive special processing
ReasonCode	reason_code	String	Code for reason for the operation that caused the event; for example, normal, autoresponse, sent, forwarded, or redirected. Set by routing strategy Stop object.
ServiceObjective	service_objective	Integer	Time objective for servicing the interaction. The contact center may define a service objective for each combination of customer segment, service type, and media type.
ServiceType	service_type	String	Code for type of service being requested; for example, `Sales`, `Service`, `Information`. Routing strategy sets this property.
Subject	subject	String	Taken from e-mail interaction

Custom Properties

The `interactions` table contains a number of customizable properties, listed in Table 26.

Note: This section does not apply to release 7.6.1; see “Flexible Configuration of Custom Properties” on page 214.

Table 26: Custom Properties

Name	Name in Interactions Table	Type
CustomString1	custom_1s	String
CustomString2	custom_2s	String
CustomString3	custom_3s	String
CustomString4	custom_4s	String

Table 26: Custom Properties (Continued)

Name	Name in Interactions Table	Type
CustomString5	custom_5s	String
CustomString6	custom_6s	String
CustomNumber1	custom_1n	Integer
CustomNumber2	custom_2n	Integer
CustomNumber3	custom_3n	Integer

To customize these properties use the following procedure.

Procedure: Using custom properties

Start of procedure

1. In Configuration Manager, open the required tenant folder and then open the `Business Attributes` folder.
2. Create a new Business Attribute:
   - Name = `InteractionCustomProperties`
   - Display name = `Interaction Custom Properties`
   - Type = `Custom`

   If such an attribute already exists go to the next step.
3. Expand `Interaction Custom Properties` and open its `Attribute values`.
4. Create an attribute value with the name of the property that you want to use (do not use spaces); for example, `AccountNumber` or `InsurancePolicyNumber`. Give it an appropriate display name (spaces are allowed); for example, `Account Number`.
5. In your new attribute value, go to the Annex tab and create a section called `translation`.
6. In the new `translation` section, create an option called `translate-to`.
7. Give the new `translate-to` option one of the custom property names from Table 26 as a value.

End of procedure

Next Steps

- You can now use the new custom property to attach data to an interaction and to define conditions and orders of Views and snapshots.

Other Properties

These properties are listed in Table 27. They are used by media servers. You cannot use these properties on the `Condition` and `Order` tabs of Views in Business Processes. You must not change them.

Note: For significant changes to these properties in release 7.6.1, see "Changes in Release 7.6.1" immediately below.

Table 27: Other Properties

Name	Type	Description
To	String	Destination e-mail address of an e-mail or web form
Header_*	String	Content of the header of an e-mail

The `To:` property is required if you use the `Chat Transcript` object in routing strategies. It supplies the value for the `To` field of the outbound e-mail that sends the chat transcript. One way to give this property a value is in a routing strategy; for details, see "Chat Transcript" under "E-Mail Objects" in Chapter 3 of the *Universal Routing 7.6 Reference Manual.*

Changes in Release 7.6.1

The following changes to the `interactions` table were made in release 7.6.1 and are relevant to the topic of this chapter.

Business Properties Not Stored as Independent Fields

Of the properties listed in "Business Properties" on page 210, only `Priority` and `ServiceObjective` are stored as independent fields of the `interactions` table in release 7.6.1. All other Business Properties are stored in the `flexible_properties` field. Here is a list: `CaseId, CategoryId, ContactId, CustomerSegment, DispositionCode, FromAddress, FromPersonal, Mailbox, ReasonCode, ServiceType, Subject.`

This means that, as delivered by Genesys, only `Priority` and `ServiceObjective` can be used on the `Condition, Order,` and `Segmentation` tabs of Views. However, if you create a custom property (see "Custom Database Fields" on page 188) corresponding to any of the other Business Properties, then that property can be used on the `Condition, Order,` and `Segmentation` tabs of Views. You can also do this for the `To` and `Header_*` properties described in "Other Properties" on page 213.

Flexible Configuration of Custom Properties

In release 7.6.1, the `interactions` table does not include the customizable properties CustomString1–6 and CustomNumber1–3 that are described in "Custom Properties" on page 211. You can still create custom properties, with greater flexibility then before, as described in "Custom Database Fields" on page 188. These can be used on the `Condition`, `Order`, and `Segmentation` tabs of Views.

Business Attributes

Business Attributes differ from interaction properties, although the two share some names (for example, `MediaType` and `Case ID`).

- Interaction properties are fields in the `interactions` table. Their main purpose is to identify and describe interactions during processing.
- Business Attributes are objects in the configuration database. They have two purposes.
  - To represent data concerning areas other than traditional telephony. Each Business Attribute contains a range of possible values for a data category. In this sense, Business Attributes are like a dictionary.
  - To convey data between software components that have no other means of communication. Examples are Category Structure and Screening Rules. These are among the Business Attributes that you must not change in Configuration Manager.

Category Structure is an example of a Business Attribute that users must not change. You create and edit categories in Knowledge Manager. The categories that you create also appear in Configuration Manager, in `Resources > Business Attributes > Category Structure`. It is possible to select a category in Configuration Manager, open its `Properties` dialog box, and make changes as you would to any configuration object. But this could create severe problems and the system would probably not operate properly when trying to handle this category. Knowledge Manager is the only safe tool to use for creating and editing categories.

Table 28 lists the Business Attributes and Business Attribute Values that are relevant to Multimedia but are not represented in the `interactions` table.

Table 28: Business Attributes That Are Not in the Interactions Table

Name	Change
CategoryStructure	Prohibited
ContactAttributes	Prohibited

Table 28: Business Attributes That Are Not in the Interactions Table (Continued)

Name	Change
ContactAttributes/AccountNumber	Allowed
ContactAttributes/LastName	Prohibited
ContactAttributes/PhoneNumber	Prohibited
ContactAttributes/PIN	Allowed
ContactAttributes/Title	Prohibited
EmailAccounts	Allowed
InteractionAttributes	Prohibited
Language	Allowed
RootInteractionID	Prohibited
ScreeningRules	Prohibited
StopProcessingReason	Allowed

Table 29 repeats three of the Business Attributes from Table 28 for which change is allowed, and adds information. Note that all three of these Business Attributes must be present for Multimedia to work properly.

Table 29: Business Attributes, Change Allowed

Name	Default Value	Comment
EmailAccounts	None	The Multimedia configuration wizard creates two values: one for the account that E-mail Server Java pulls e-mails from in order to bring them into the Multimedia system, and one for an external agent. You may want to use Configuration Manager to create additional values, for example if you want additional e-mail addresses to be available for `Forward` and `Redirect` objects in routing strategies. Configure the address for this value on the `Annex` tab > `General` section > `address` option. The address must comply with RFC2822 and therefore it must be encoded according to RFC2047. Examples: legal@mycompany.com ""Legal Dpt"" <legal@mycompany.com> Type = custom.
Language	English	Used by Knowledge Manager (along with Tenant) to group category trees. You must add the value `unknown` if you want to train a model that classifies by language. See "Training" on page 84 and "Notes on Language" on page 115.
StopProcessingReason	AutoResponded Forwarded Normal Re-directed Sent Terminated	Specified by agent application when it stops the interaction.Must be selected in the IRD `Stop E-mail` object. **Note:** You must manually add the value `Abandoned`, which is required by Chat Server.

Translations

In IRD, on the `Conditions` and `Order` tabs of a View object in an Business Process, you can write statements that refer to the three types of interaction properties that are represented as independent fields in the `interactions` table: system, business, and custom. You can also use translations, which provide database manipulation tools that are independent of the underlying database.

Translations enable you to use the interaction property names as presented in this chapter rather than database table field names. They provide a unified representation, regardless of the underlying database, of a collection of functions. Some deal with time and date, others with attributes of the specific interaction; for example, `_age()` calculates the interaction's age in seconds, regardless of database type used.

Notes: Bear in mind the following:

- Translation does not hide the original database syntax. Interaction Server translates whatever it can, leaves the rest unchanged, and transmits it all to the database. Therefore you can also use database field names and any database-specific constructs on the `Conditions` and `Order` tabs.
- Many of these translations are time sensitive. As such they have a dependency on the `freeze-interval` option of the Interaction Queue View object. For details, see the discussion of this option in the "Interaction Server Options" section of the "Configuration Options" chapter of the *Multimedia 7.6 Reference Manual.*

_timestamp

Specifies date constants in conditions. The single argument is a character constant that represents a date in the common form used by Interaction Server.

Usage

```
_timestamp('yyyy-mm-ddThh:mi:ssZ')
```

OR

```
_timestamp('yyyy-mm-dd hh:mi:ss')
```

Translations Performed by Database

For Oracle 9:

```
TO_DATE('yyyy-mm-dd hh:mi:ss', 'YYYY-MM-DD HH24:MI:SS')
```

For Microsoft SQL:

```
CONVERT(DATETIME, '2003-11-26 17:43:23', 102)
```

For DB2:

```
TIMESTAMP('yyyy-mm-dd hh:mm:ss')
```

_current_time

Calculates the current UTC date-time.

Use this function to avoid the confusion caused by mixing UTC and local times. Since Interaction Server manipulates only UTC time, and all dates in the

database are saved as UTC time, there is no standard function to get the current local date-time.

Database-specific functions can also be used to get the local time, such as Microsoft SQL's `getdate()` or Oracle's `sysdate`.

Usage

```
_current_time()
```

Translations Performed by Database

For Oracle 9:

```
cast(SYS_EXTRACT_UTC(SYSTIMESTAMP) as date)
```

For Microsoft SQL:

```
GETUTCDATE()
```

For DB2:

```
(CURRENT TIMESTAMP - CURRENT TIMEZONE)
```

_timestampdiff

Calculates the difference, in seconds, between two timestamps.

Both arguments should be of type timestamp, which differs depending on the database engine:

- Oracle—date
- Microsoft SQL—datetime
- DB2—timestamp

Usage

```
_timestampdiff(timestamp date1, timestamp date2)
```

Translations Performed by Database

For Oracle 9:

```
(((date1) - (date2))*86400)
```

For Microsoft SQL:

```
cast(cast(((date1) - (date2)) as float)*86400 as integer)
```

For DB2:

```
timestampdiff( 2, char((date1) - (date2)) )
```

_timestampadd

Method of increasing and decreasing dates. Generates a date increased or decreased by a specified number of seconds. The second parameter can be negative, to make the result a date earlier than the original date.

Usage

```
_timestampadd(timestamp date, integer seconds)
```

Translations Performed by Database

For Oracle 9:

```
(date + (seconds)/86400)
```

For Microsoft SQL:

```
dateadd(second, seconds, date)
```

For DB2:

```
(date + (seconds) second)
```

_age

Calculates the age in seconds of the interaction; that is, the difference between the current time in UTC and the interaction attribute `ReceivedAt`.

Usage

```
_age()
```

Translations Performed by Database

For Oracle 9:

```
((cast(SYS_EXTRACT_UTC(SYSTIMESTAMP) as date) - received_at)*86400)
```

For Microsoft SQL:

```
cast(cast((getutcdate() - received_at) as float)*86400 as integer)
```

For DB2:

```
timestampdiff( 2, char((current timestamp - current timezone) -
received_at))
```

_time_in_queue

Calculates the time, in seconds, that the interaction has spent in the queue; that is, the difference between the current time in UTC and the attribute `PlacedInQueueAt`.

Usage

```
_time_in_queue()
```

Translations Performed by Database

For Oracle 9:

```
((cast(SYS_EXTRACT_UTC(SYSTIMESTAMP) as date) -
placed_in_queue_at)*86400)
```

For Microsoft SQL:

```
cast(cast((getutcdate() - placed_in_queue_at) as float)*86400 as
integer)
```

For DB2:

```
timestampdiff( 2, char((current timestamp - current timezone) -
placed_in_queue_at))
```

_time_in_same_queue

Same as `_time_in_queue` except that it is based on the `MovedToQueueAt` property, which is updated only when the interaction is placed into a queue that it has not been in before.

Provides a unified method of calculating the amount of time that the interaction has spent in the current queue. It calculates the difference in seconds between the current time in UTC and the attribute `MovedToQueueAt`.

Usage

```
_time_in_same_queue()
```

Translations Performed by Database

For Oracle 9:

```
((cast(SYS_EXTRACT_UTC(SYSTIMESTAMP) as date) -
moved_to_queue_at)*86400)
```

For Microsoft SQL:

```
cast(cast((getutcdate() - moved_to_queue_at) as float)*86400 as
integer)
```

For DB2:

```
timestampdiff( 2, char((current timestamp - current timezone) -
moved_to_queue_at))
```

_empty

Usage

```
_empty(string value)
```

The result is a logical expression that is true if the string value is either null or an empty string, and is false otherwise.

Translations Performed by Database

For Oracle:

```
((value) is null)
```

For all other database platforms:

```
(((value) is null) or ((value)=''))
```

Note: The reason the Oracle condition is different is that Oracle interprets an empty string as null, and a condition like `(anything=null)` is always false.

_not_empty

Usage

```
_not_empty(string value)
```

The result is a logical expression that is false if the string value is either null or an empty string, and is true otherwise.

Translations Performed by Database

For Oracle:

```
((value) is not null)
```

For all other database platforms:

```
(((value) is not null) and (not ((value)='')))
```

Chapter

8 Transitioning to Multimedia from ICS 6.x

This chapter describes the transition from Internet Contact Solution (ICS) 6.x to Multimedia 7.x. It includes the following topics:

- Overview, page 223
- Converting the Contact Server Database, page 224
- Transition Results, page 233
- Other 6.5.x Objects, page 238

Overview

Multimedia 7.x has much of the same functionality as Internet Contact Suite (ICS) 6.x. However, Multimedia does not replace ICS. Similarly, there is no migration of ICS components to Multimedia components. You cannot migrate ICS configuration data to 7.0. You must configure and install a new Multimedia solution from scratch.

There is one aspect of ICS 6.x data that you can convert for use in Multimedia: the data stored in and handled by the Contact Server database (with a few exceptions). The Multimedia equivalent of Contact Server is Universal Contact Server. A transition tool, supplied with Multimedia 7.x, can convert Contact Server data from ICS 6.5.1 to Multimedia 7.x. This tool is a standalone, GUI-less Java utility, delivered as part of the Multimedia installation package.

Transition Strategy

If you have ICS 6.0, 6.1, or 6.5.0, you must first migrate your Contact Server database to ICS 6.5.1, then use the transition tool to convert the database for use in Multimedia 7.x. As a third step, you can then re-create the items that cannot be converted.

To summarize, ICS 6.x objects fall into three groups in terms of transition:

- Contact and interaction records in the Contact Server database are converted with little or no change. Converting the database with the transition tool is described in "Converting the Contact Server Database" on this page.
- Other Contact Server database objects are converted with some changes due to differences between ICS 6.x and Multimedia 7.x. These include standard responses and Content Analyzer rules (match criteria). They are described in "Transition Results" on page 233.
- All other objects cannot be converted or reused, but must be re-created. These include
  - Contact Server database objects such as Content Analysis training models
  - Other objects such as strategies and event handlers.
  - Any objects created as the result of customizing the ICS database.

  Suggestions for re-creating some of these are described in "Other 6.5.x Objects" on page 238.

Converting the Contact Server Database

In general outline, the database transition process is as shown in Figure 85.

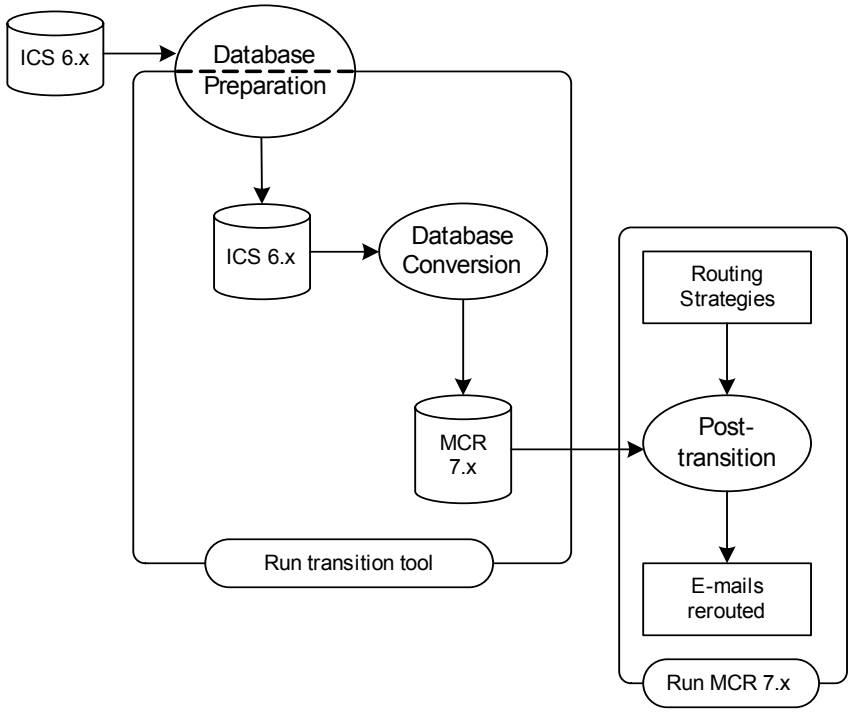


Figure 85: Outline of Transition Process

Table 30 lists the four overall stages in the database transition process, and provides links to the detailed description of the procedure for each stage.

Table 30: Overall Procedure for Database Transition

Stage	Description	Procedure
1. Preliminaries	Configure required objects and create required routing strategy or strategies.	"Preliminaries to database transition" on page 225
2. Preparation	Run two SQL scripts on the ICS 6.x database to prepare it for conversion.	"Preparing the database" on page 228
3. Conversion	Run the transition tool.	"Running the Transition Tool" on page 229
4. Post-transition	Start your Multimedia 7.x system. It uses the routing strategy you created in Step 1 to process any e-mails that were pending in the ICS 6.x database.	No separate description

Conversion Procedures

Procedure: Preliminaries to database transition

Purpose: Configure required objects, if they do not exist already.

Start of procedure

1. Install Genesys Framework 7, including at least the following:
   - Configuration Server 7
   - Multimedia Third Party Components
   - Multimedia Universal Contact Server
   - Multimedia Universal Contact Server Transition Tool
   - Multimedia E-mail Server Java
2. Configure at least the following objects:
   - One Tenant.
   - Universal Contact Server (UCS), specifying the Tenant just described.
   - A Contact Server 6.5.1 Application object. This object must replicate your ICS 6.x Contact Server object: in particular, you must copy all options from your ICS 6.x Contact Server to this application. You can use the import and export functionality of Configuration Manager to transfer the option values.

- E-mail Server Java, with a connection to UCS. This enables the system to reprocess e-mails which are pending in ICS 6.x.

Note: Do not run E-mail Server Java until the transition process is complete.

- All Persons that exist in your ICS 6.x configuration, with identical user names. This enables the system to convert the owner history of interactions. Persons are relevant to data conversion in the following ways:
  - Some data objects, such as agents' personal spelling dictionaries and agent's favorite standard responses, can only be converted if the Person identified as their Owner exists in the 7.x environment.
  - Some unfinished interactions are assigned to an agent in the ICS 6.x database (other unfinished interactions, such as those that have not been routed, may not be assigned). The transition tool can convert these interactions so that a post-transition routing strategy (described in Step 4) can assign them to the same agent. But such conversion can succeed only if both (a) the transition tool command line does not include the parameter `-dontcheckperson` (described on page 230) and (b) the agent exists as a Person object in the 7.x environment.
  - Converted interactions include an Owner attribute that identifies the last agent who updated the interaction record. If this agent does not exist as a Person in the 7.x environment, the interaction is still converted, but the Owner attribute is blank.

3. Make a note of the following information, which you will need for the command-line arguments of the transition tool:
   - Host and port of Configuration Server
   - Name of the Contact Server 6.5.1 Application object
   - Name of the UCS Application object
   - Name of the E-mail Server Java Application object
4. Create and load a routing strategy to handle pending e-mails. In deciding how to configure this strategy, you should be aware of the following:

   After data conversion, E-mail Server Java 7.x submits to Interaction Server all e-mails which were pending in ICS 6.x. The queues that it submits them to are specified as follows:
   - Pending outbound e-mails go to the queue specified in the transition tool's `-outboundqueue` command-line parameter, which is described on page 230.
   - Pending inbound e-mails go to a queue defined as follows:
     - If the e-mail has a field `EmailIn.Mailbox` equal to the name of a `pop-client` section (in the options of the E-mail Server Java Application object in the Configuration Layer 7.x), E-mail Server

Java submits the e-mail to the queue that is defined by the endpoint option managed in this `pop-client` section (see the *Multimedia 7.6 Reference Manual* for a description of endpoints).

- If there is no endpoint configured, or if the inbound e-mail has a field `EmailIn.Mailbox` not equal to the name of a `pop-client` section, (because the pop-client sections differ in the 6.5 and 7.x configurations), E-mail Server Java works in 7.1 compatibility mode. In this mode, it submits the e-mail to the queue defined by the `email-processing` section's `default-inbound-queue` option.

Also, in ICS 6.x, outbound e-mails can be of various types, such as *waiting for QA review, saved as draft by an agent, transferred to another location,* and so on. But Multimedia 7.x merges all of these types. So whatever strategy is loaded on the queue specified by the `-outboundqueue` command-line parameter (see Table 31 on page 229) treats all e-mails in that queue identically. If the strategy simply sends, it sends all of these pending outbound e-mails to customers, even those that were marked in ICS 6.x as drafts or as waiting for QA review. Because of this, Genesys recommends that you configure this strategy to route these pending e-mails to agents.

5. Plan and execute the preparation of your ICS 6.x database, as described in the following section.

End of procedure

Next Steps

- Review the limitations and restrictions described in the following section.
- Go on to prepare the database (page 228).

Limitations and Restrictions

Be aware of these limitations:

- Any customized columns that you have added to your ICS 6.5.x database cannot be converted. They will be lost.
- Once you begin the overall transition procedure, Genesys recommends that you make no configuration changes to either the ICS 6.5.x or the Multimedia 7.x environment. You are particularly discouraged from making any changes to Knowledge Management objects (Multimedia 7.x, using Knowledge Manager) or standard responses/categories/content analysis rules (ICS 6.5.x, using Content Analyzer and Response Manager).
- Genesys recommends that your UCS application not be connected (using its Connection tab in Configuration Manager) to StatServer. This prevents the system from creating inconsistent initial reporting data based on objects processed in ICS and converted to Multimedia.

Procedure: Preparing the database

Summary

The data conversion process is incremental: you can run the tool multiple times and each time it resumes the job where it left off. You can safely stop the transition tool using `Control-C`.

Note: Genesys strongly recommends against running pruning or archiving in between multiple conversion sessions. Use UCS Manager to ensure that no pruning or archiving is scheduled before transition is complete.

Start of procedure

1. Take your ICS 6.x system offline before you run the transition tool. There are two ways to accomplish this:
   - Maintenance windows—designate windows of time during which you can take your production ICS offline and run the transition tool on the database. This mode requires more time.
   - Database copy—copy your ICS database, then run the transition tool on the copy. This mode takes less time but requires more disk space for the duplicate database.
2. Locate the two SQL scripts that you will run on your ICS 6.x database before running the transition tool. These scripts are supplied, in either MSSQL or Oracle versions, in `GCTI\MCR 7.6\Universal Contact Server Transition Tool\sql_scripts\<database_type>`, where `<database_type>` is either `mssql` or `oracle`.
3. Proceed with either the maintenance windows or database copy method, as follows:
   a. Maintenance Windows
      i. Take your production ICS database offline.
      ii. Optionally, make a backup copy of your ICS database.
      iii. Run the preparation script `PrepareICSDBForTransition_<database_type>.sql` on the ICS database.
      iv. Run the population script `PopulateICSDBForTransition_<database_type>.sql` on the ICS database.
      v. Run the transition tool on the ICS database. See “Running the Transition Tool” on page 229 for details.
      vi. When the maintenance window is over, stop the transition tool using `Control-C`.

vii. Bring your production ICS database back online.

viii. At the next maintenance window, take your production ICS database offline and repeat Steps 1, 6, and 7.

b. Database Copy

i. Take your production ICS database off line.

ii. Optionally, make a backup copy of your ICS database.

iii. Make a working copy of the ICS database.

iv. Run the preparation script `PrepareICSDBForTransition_<database_type>.sql` on both the original ICS database and the copy.

v. Put the original database back on line.

vi. Run the population script `PopulateICSDBForTransition_<database_type>.sql` on the copy.

vii. Run the transition tool on the copy. See "Running the Transition Tool" on page 229 for details.

viii. When the transition is complete, take the original database off line.

ix. Run the transition tool on the original (production) database. This converts only the records created, updated, or deleted since the original database was put back on line (in Step e).

End of procedure

Procedure: Running the Transition Tool

Start of procedure

1. Run the tool directly from the shell, on the target platform operating system. Use the command `UcsTransTool` with the parameters listed in Table 31.

Table 31: Transition Tool Command-Line Parameters

Name	Required?	Values
-app	Yes	Name of your ICS 6.x Contact Server Application object
-host	Yes	Host of your Configuration Server 7.x
-port	No	Port number of your Configuration Server 7.x. The default is 2020.
-ucsapp	Yes	Name of your UCS 7.x Application object
-esjapp	Yes	Name of your E-mail Server Java 7.x Application object

Table 31: Transition Tool Command-Line Parameters (Continued)

Name	Required?	Values
-user	Yes	Name of an existing Person object. It is used for the value of the Owner attribute of categories and standard responses.
-language	Yes	The language associated with standard responses and screening rules in Multimedia 7.x.
-archive	No	No values. If this parameter is not present, data conversion applies to the Main database. If this parameter is present, data conversion applies to the Archive database. Requires an Archive database in ICS 6.x and an Archive DAP in Multimedia 7.x.
-addresscheck	No	No values. If this parameter is present, the transition tool checks the addresses of e-mails for compliance with RFC 822 and does not convert noncompliant e-mails. If this parameter is not present, the transition tool does convert noncompliant e-mails.
-outboundqueue	Yes	Name of an existing queue. After the database transition, E-mail Server Java takes any outbound e-mails that are in a pending status in the ICS 6.x database and submits them to the queue specified by this parameter.
-iKnow	No	No values. If this parameter is present, the transition tool converts category feedback and creates an extra category to associate with each standard response. See “Standard Responses, Categories, Feedback” on page 234 for full description.
-tenant	Yes if UCS has multiple tenants	Tenant name. If UCS is associated with multiple tenants, this parameter selects one of them.
-dontcheckperson	No	No values. This parameter applies to interactions that are unfinished and are assigned to an agent (see also page 226). If this parameter is present, the transition tool converts these interactions without checking that the assigned agent exists as a Person object in the 7.x environment. If this parameter is not present, the transition tool does check and converts the interaction only if it finds the Person.

Example command line:

```
UcsTransTool -app ContServ651 -host Multimedia7host -port 7070
-ucsapp UCS701 -esjapp EmailSrvJav701 -user admin -language English
-outboundqueue TransitionQueue
```

Note: Genesys Technical Support may instruct you to use additional parameters for debugging purposes.

2. Upon launching, the tool displays the following on its console:
   - Number of records to convert per object type
   - Total number of records to convert
3. Then the tool starts to convert records, sequentially type after type, and for each record type being converted, it displays the following progress information:
   - Number of records converted
   - Predicted end time for this type of object
4. When all records for a given object type have been converted, it displays:
   - Number of records successfully converted
   - Number of errors encountered
   - Time taken
   - Average throughput

End of procedure

Next Steps

- Restart your Multimedia 7.x system. The system takes the routing strategy that you created in Step 4 on page 226 and uses it to process any e-mails that were pending in the ICS 6.x database.

Corruption Recovery

While running, the transition tool checks for corruption in the ICS 6.x database. If it detects corruption, it makes repairs on the corrupt data, with the repairs taking effect in the new Multimedia 7.x database. The tool gives no indication of this process.

Table 32 shows the types of corruption and the correction that the transition tool makes on them.

Table 32: Corruption Types and Corrections

Type of Corruption	Situation in ICS	Correction in Multimedia
Threading of interactions	The interaction has no parent interaction or an invalid parent interaction.	Set the interaction as the root interaction of a new thread of discussion.
Contact attributes	Of multiple `EmailAddress` or `PhoneNumber` attributes, there is either no primary attribute or multiple primary attributes.	Randomly select one and set it as primary.

Table 32: Corruption Types and Corrections (Continued)

Type of Corruption	Situation in ICS	Correction in Multimedia
Interaction contact	No contact is defined for an interaction or the contact is not available.	Use interaction data to create a new contact: • EmailIn: use e-mail's From address. • EmailOut: use e-mail's To address. • WebForm: use e-mail address. • PhoneCall: use phone number.
Interaction.DateSent	The DateSent field of the interaction is in the future.	Use value of DateCreated.
Value of isDone field is contradictory with other fields	isDone = true (interaction is finished) but Status = 1 (interaction is a draft assigned to an agent)	Convert Status to Pending. The interaction will be processed by a post-transition strategy.
	isDone = false (not finished) but the interaction is an automated response which has already been sent.	Convert Status to Stopped.

Errors and Recovery

If an error occurs, you can find a list of IDs of records that failed conversion in the file `log/UcsTransToolObjectsErrorList.log`.

The following are possible causes of errors during the conversion process:

1. Consistency error in the ICS 6.x record (for example, an invalid reference to a dependent object).
2. Format error in the ICS 6.x record (for example, an unknown charset or an unparsable email address).
3. System error (for example, no access to the ICS 6.x database or to UCS, or disk full).
4. Defect in UCS, the UCS API, the ICC API, or the transition tool itself.

To recover from (1) and (2), manually correct the ICS 6.x records, then run the transition tool again.

To recover from (3), fix the system error, then run the transition tool again.

To recover from (4), upgrade the defective component, then run the transition tool again.

After Transition

Inspect your database both before and after transition to check that no unwanted modifications are introduced by the transition process.

Transition Results

As mentioned on page 224, the transition tool converts contact and history data with little or no change. This is also the case with field codes.

Other objects in the Contact Server database undergo some change as the result of conversion. This section describes some differences between the pre- and post-conversion versions of these objects.

All of the objects described in this section acquire the following attributes upon conversion:

- Tenant—the first tenant in the list on the `Tenants` tab of the UCS Application object.
- Language—the value of the `-language` parameter in the transition tool command line.

The rest of this section presents information about individual object types.

Content Analysis Rules: Match Criteria (Multimedia Screening Rules)

ICS 6.x Content Analysis rules cover several functionalities that are redistributed in Multimedia 7.x. In terms of the four tabs of ICS Content Analyzer, the distribution is as follows:

- The functionalities located on the `Match Criteria` tab correspond to Multimedia 7.x *screening rules,* controlled in Knowledge Manager and described in “Screening Rules” on page 58.
- The functionalities located on the `Routing Properties`, `Automated Actions`, and `E-Mail Properties` tabs are taken over by strategy objects in Multimedia 7.x.

The transition tool converts match criteria to screening rules, with the following changes:

- Any match criteria that contain the binary `NOT` operator (for example, `A NOT B`) are changed to use `AND NOT`. For example, `A NOT B` converts to `A AND NOT B`.
- Any special characters in the name of the Content Analysis rule are converted to underscore (_). For example, the name `Thanks&Redirect` would become `Thanks_Redirect`.

Content Analysis rule functionalities other than match criteria must be re-created in routing strategies. See also “Content Analysis Rules” on page 238.

Standard Responses, Categories, Feedback

Background

- In ICS 6.x, each standard response is associated with at least one category. You can create a standard response without a category but Response Manager assigns such standard responses to a category called Uncategorized.
- In ICS 6.x, the relation between standard responses and categories is many-to-many: one standard response can belong to many categories, and one category may have many standard responses. In Multimedia 7.x the relation between standard responses and categories is many-to-one: one standard response can belong to only one category, although one category may have many standard responses. This difference between ICS 6.x and Multimedia 7.x has consequences for conversion, as explained below.
- Feedback is data that records a relation between an object (either a category or a standard response) and a interaction, signifying that object O is a good match with interaction I. In ICS O is a standard response; in Multimedia O is a category. This difference also has consequences for conversion.

Name Conversion

As described on page 26, names of Knowledge Manager objects (categories, standard responses, and so on) in Multimedia 7.x must consist only of alphanumeric characters (A–Z, a–z, 0–9), hyphen (-), underscore (_), and space (names must also be no more than 64 characters long). If any ICS 6.x standard response or category has a name that violates this rule, the transition tool replaces the nonconforming characters according to the correspondences shown in Table 33.

Table 33: Replacement in Knowledge Manager Names

Original	Replacement	Original	Replacement
~	0	?	F
!	1	/	_
@	2	,	H
#	3	.	I
$	4	`	J
%	5	{	K
^	6	}	L

Table 33: Replacement in Knowledge Manager Names (Continued)

Original	Replacement	Original	Replacement
&	7	[	M
*	8	]	N
(	9	|	O
)	A	:	P
+	B	;	Q
=	C	"	R
<	D	'	S
>	E	\	T

Transition Process

The transition tool converts standard responses, categories, and feedback as follows (these numbered steps do not necessarily happen in chronological order):

1. It creates a root category named `ICS Migrated <appname> <langname>`, where `<appname>` is the value of the `-app` parameter and `<langname>` is the value of the `-language` parameter in the transition tool command line. An example might be `ICS Migrated myContactServer German`.
2. All categories existing in ICS 6.x become subcategories of the root category.
3. Standard responses remain associated with their categories. But any standard response that was associated with multiple categories becomes a set of duplicates, each associated with one category. Consider an example with three categories, called Cat1 Cat2 Cat3, and three standard responses, called SR1 SR2 SR3, associated as shown in Figure 86.

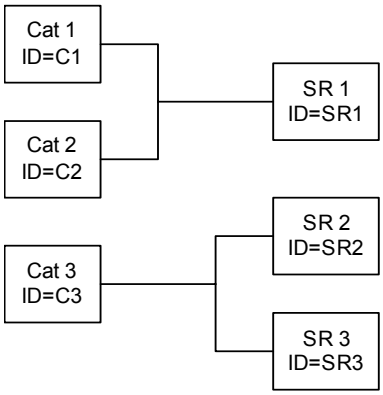


Figure 86: Example Standard Responses and Categories Before Conversion

Figure 87 shows the same categories and standard responses after conversion.

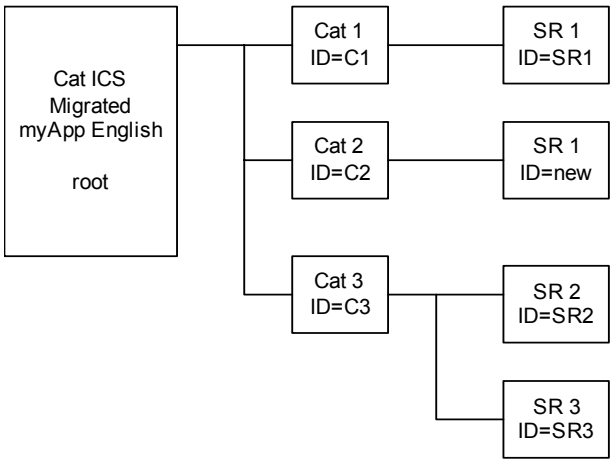


Figure 87: Example After Conversion

Notice SR 1. Before conversion it is a single standard response assigned to two categories. Because Multimedia 7.x does not support this type of relation, the transition tool replaces it with two identical standard responses (differing only in their database IDs), each assigned to a single category.

In this scenario, feedback is not converted. To convert feedback, you must include the parameter `-iknow` in the transition tool command line.

4. With the `-iknow` parameter, the output of the transition tool is different: the tool creates an additional layer of categories so that each standard response is associated with exactly one category. Figure 88 shows the results of converting the same example as Figure 86, but with the `-iknow` parameter.

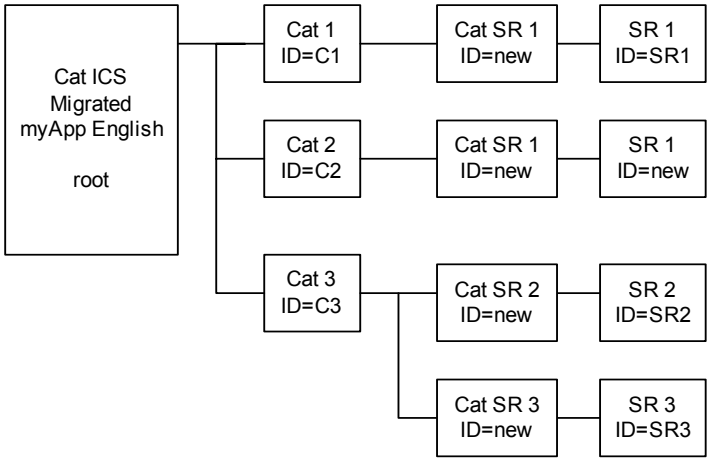


Figure 88: Example After Conversion, with the -iKnow Parameter

Notice SR 2 and 3. Before conversion, both belong to Cat 3. After conversion, each belongs to its own category, and it is these lower-level categories that both belong to Cat 3.

The advantage of this is as follows. Before conversion, feedback data could mark interactions as being good matches for SR2 and SR3 separately. After conversion, feedback data must mark interactions as good matches for categories, not standard responses—but before conversion, SR2 and SR3 belong to the single category Cat3. Taking the feedback data for SR2 and SR3 and merging it as feedback for Cat3 would lose the distinction between SR2 and SR3. Using the `-iknow` parameter adds a level of categories that have one-to-one relations with standard responses. Associating feedback data with this level of categories preserves the distinctions between standard responses like SR2 and SR3.

Other attributes of the converted standard responses are as follows:

- Name and Description—the name of the ICS 6.x standard response is copied into both of these attributes.
- Usage type—all three types are selected and specified as `Not Active`.
- Status—Approved.
- Owner—the value of the `-user` parameter in the transition tool command line.
- Date modified—date that the transition tool is run.
- Expiration date—empty.
- Version—0.

Field Codes

Field codes simply transfer their names and contents from ICS 6.x to Multimedia 7.x.

Other 6.5.x Objects

Objects that you created or customized in ICS 6.x cannot be directly migrated or transferred to Multimedia 7.x. This section lists some of the ICS 6.x objects and discusses what you can do to transition them to Multimedia 7.x.

Contact Center Objects

DNs

Unlike ICS 6.x, Multimedia 7.x does not use DNs that are specified for particular nonvoice media such as e-mail and chat. You can delete any such DNs from your configuration.

Places

Since DNs no longer serve to mediate the flow of interactions to agents, you must configure agent capacity rules. You can do this for either a tenant (the default) or individual agents.

Content Analysis Rules

These ICS 6.x rules have two general functions: scan an incoming message to see if it matches certain criteria, and perform specified actions if there is a match.

As stated in "Content Analysis Rules: Match Criteria (Multimedia Screening Rules)" on page 233, the transition tool converts match criteria to screening rules.

However, the actions that were performed by the `Routing Properties` and `Automated Actions` tabs of ICS Content Analyzer are taken over by strategy objects in Multimedia 7.x. For example, Content Analyzer included an `Intelligently analyze the content of the message` check box. In Multimedia, this function is performed by the `Classify` and `Classify segmentation` strategy objects. See *Universal Routing 7.6 Reference Manual* for details on these and other IRD e-mail objects.

Strategies

There is no simple way to take routing strategies that you created for ICS 6.x and use them with Multimedia 7.x.

The best way to understand the relation between ICS 6.x and Multimedia 7.x is to study the Interaction Workflow Samples, a component of Multimedia 7.x. These exemplify many of the tasks that users commonly require (for example, sending interactions to QA review). If you have these or similar tasks set up in

your ICS 6.x environment, you should compare your setup with the analogous part of the Interaction Workflow Samples.

Note also the following:

- In Multimedia 7.x, strategies must be contained in Business Processes. Business Processes are created and edited using Interaction Routing Designer; see *Universal Routing 7.6 Interaction Routing Designer Help* for details. If you want to use strategies from your ICS 6.x in Multimedia 7.x, you must embed the strategies in one or more interaction workflows.
- Universal Routing 7.0.1 contains many more e-mail routing objects than previous versions. Also some objects (such as `Target`) that were present in previous versions have significant differences in 7.0.1. See *Universal Routing 7.6 Reference Manual* for descriptions.

Other Objects

Event Handlers

Multimedia 7.x has no equivalent of custom event handlers.

Callback Records

The transition tool converts ICS 6.x callback records, so they are available in the Multimedia 7.x UCS database. But the callback media is not supported in Multimedia 7.x, with the following consequences:

- The transition tool converts records of all callbacks, including unfinished ones. Unfinished callback are converted as finished. They are not reprocessed in Multimedia 7.x.
- There will be no new callback records. The function of ICS Callback Server 6.x is performed by Genesys Voice Callback 7.0, which has its own storage system for callback requests. For details see the *Voice Callback 7 Getting Started Guide* and the resources that it cites.

iKnow Training Models

Classification Server (part of the optional iKnow package in 6.5.x) now classifies on Category objects rather than Standard Response objects as it did in 6.x. Because of this fundamental change, Classification Server 7.x cannot use any models created in ICS 6.x.

Web Sample and Web API

ICS 6.x Web Sample (WebStarterApp in older releases) cannot be migrated or converted at all. Web API Server likewise cannot be migrated or converted.

Glossary

This glossary provides an alphabetical listing of terms, names, and concepts used in the documentation for Genesys Multimedia 7.6. This glossary is not divided into sections.

Note: Items that apply only to Genesys Content Analyzer have definitions that begin with *(Genesys Content Analyzer)*.

accuracy

(Genesys Content Analyzer) A general term for how correct a model is in assigning text objects to categories. If you produce a model using cross-validation or test it on a training object, you obtain specific accuracy ratings called precision and recall. Contrast confidence, which is part of what a model does when assigning a text objects to categories: it produces a list of categories with a confidence level indicating the model's assessment of how likely it is that this text object belongs to this category.

acknowledgment

A possible use for standard responses: a message sent automatically to inform the sender that the message has been received.

agent capacity rule

Rule specifying the maximum number of interactions of each media type an agent can handle. Agent capacity rules are used in routing; if an agent is up to capacity for a given media type, the strategy may still send that agent further interactions of other media types. An agent may have different settings for different media types; for example, a capacity rule may classify the agent as busy for chat interactions (already engaged in chat) but not for e-mail interactions (may be sent additional e-mail interactions). You can configure agent capacity rules at various levels of generality (Agent, Tenant) in Configuration Layer. Configure using the Agent Capacity Wizard.

archiving, pruning

Means of keeping the size of the Universal Contact Server database under control. Archived records are removed from the Main database and stored in the Archive database. Pruned records are deleted and not stored.

autoresponse

A possible use for standard responses: a message sent as an automated response to an incoming e-mail.

business process

In Interaction Routing Designer (IRD), a set of objects (mainly queues and routing strategies) that are available for use in constructing interaction workflows. Objects in business processes function in an interaction workflow only if you have placed them into a workflow and connected them to at least one other object in the workflow.

cache

Database associated with Interaction Server that stores transient information about interactions. This information includes:

- operational data.
- queues through which the interaction passes during processing.

This cache has sometimes also been called Persistent Queue, Interaction Cache, or iCache.

category

An item in a system of categories and subcategories, called a category tree, created and edited using Knowledge Manager. A category, besides denoting a concept, may be associated with one or more standard responses and one or more screening rules. If an incoming e-mail is assigned to the category, the category's standard responses can serve either as the content for an automated reply or as suggestions for agents to use in their replies. A terminal or leaf category is one that contains no subcategories. A nonterminal category is one that does have subcategories.

co-browsing

The ability for agents and customers to simultaneously navigate shared web pages; sometimes also known as web collaboration or conavigation. In release 7.6 this functionality is provided by two components, Genesys Co-Browsing Server and KANA Response Live Server. These components are provided as Genesys Web Collaboration, which is an option that you can add to either Genesys Web Media or Genesys Inbound Voice. See also "Multimedia and the CIM Platform" on page 10.

confidence

(Genesys Content Analyzer) A numerical score, ranging from 1 to 100, indicating the likelihood, according to a particular model, that a text object belongs in a certain category. You set confidence level as one of the attributes of the IRD objects `Classify` and `Classification switch`. Contrast accuracy, which is an assessment, produced by testing, of the correctness of a model's assignment of text objects to categories. Confidence expresses a model's guess about a categorization; accuracy rates the correctness of that guess.

content analysis

(Genesys Content Analyzer) What Content Analyzer does: applies natural language-processing technology to analyze the content of incoming interactions. *See also* Genesys Content Analyzer and model.

e-mail

Knowledge Manager uses this term, interchangeably with *message,* to refer to interactions that it applies screening rules or content analysis to. Although most interactions that are screened or classified are expected to be e-mail messages, in fact these operations can apply to any interaction that has text somehow associated with it. The text can be the body of the interaction (e-mail, chat), or it can be more obliquely associated with it (as user data, for example).

ESP

External Services Protocol. Used by Interaction Server to communicate with servers that perform a specific service when requested to do so. Such servers are called ESP servers. Classification Server is an ESP server, as is E-mail Server Java when it generates an acknowledgment or autoresponse (when E-mail Server Java processes incoming or outgoing e-mails, it is a media server). You can create custom ESP servers using the Genesys Open Media Platform SDK.

field code

Formula using variables, constants, and operators; allows a standard response to be personalized relative to the contact it is addressed to. For example, a response beginning `Dear <$Contact.FirstName$>` may be sent to dozens of recipients; in each message *`<$Contact.FirstName$>`* is replaced by the first name for that contact as listed in the Universal Contact Server database. This process of replacement is called rendering.

Genesys Content Analyzer

Optional extension of Genesys Knowledge Management, activated by presence of the proper license key. Content Analyzer uses natural language-processing

technology to scan incoming e-mails, assigning the e-mail to one or more categories with a percentage confidence rating. The category assignments can then be used to pull suggested responses from the Standard Response Library. Content Analyzer creates and refines its recognition algorithms by training.

interaction

In the broad sense: an attempted communication between a customer and a contact center, in either direction. The attempt may be successful or not; it has a media type; it may give rise to other interactions (as when an incoming e-mail gives rise to an automatic acknowledgement). It may belong to a series of related interactions, known as a *thread*.

In the narrow sense: a software object, created by a server, that represents an interaction in the broad sense.

interaction workflow

Created by IRD. Specifies the high-level flow of interactions between various contact center objects (mainly queues and routing strategies). Must begin with a single queue for inbound interactions. Normally uses the contact center objects belonging to one business process but may include objects from more than one. Any switching between business processes takes place within routing strategies.

IRD

Interaction Routing Designer; GUI component of Genesys Universal Routing used to design routing strategies that handle interactions as well as interaction workflows that encompass routing strategies, queues, and other objects.

media server

Component that interfaces with a particular media to bring interactions into the Genesys Multimedia system. Supported media are e-mail (E-mail Server Java) and chat (Chat Server).

message

See e-mail.

model

(Genesys Content Analyzer) classification model; a resource that Classification Server uses to classify e-mails. A model is associated with a category structure; it contains a statistical representation of each category in the structure. To classify a new e-mail message Classification Server compares it with the representation of each category. Then the server returns a list of categories, each with a percentage rating of the confidence with which the e-

mail in question can be assigned to that category. Model files are created and refined by training and are stored in the Universal Contact Server database.

natural language processing

(Genesys Content Analyzer) technology that operates on data that is in a language used by humans (as opposed to programming or other machine languages).

online

An online interaction is one that takes place online in real time. For example, chat is online; e-mail is not.

operational data

What the media server sends to Interaction Server for each incoming interaction: interaction ID, originating party, time received, and so on.

precision

(Genesys Content Analyzer) One mathematical expression of a model's accuracy. Given the following for a category X:

a = the number of items the model correctly assigns to X

b = the number of items the model incorrectly assigns to X

Then *precision* = $a/(a + b)$

See also "Average Results Subtab" on page 130.

pruning

See archiving, pruning.

queue

Or interaction queue: in Multimedia, a logical entity in the Interaction Server cache database. Typically, there is an Inbound queue, an Outbound queue, and various intermediary queues.

recall

(Genesys Content Analyzer) One mathematical expression of a model's accuracy. Given the following for a category X:

a = the number of items the model correctly assigns to X

c = the number of items the model incorrectly rejects from X (that is, items that the model should assign to X but does not)

Then *recall* = $a/(a + c)$

See also "Average Results Subtab" on page 130.

rendering

The process of taking a formula contained in a field code, performing the operation described by the formula, and substituting the result of the operation for the formula text.

routing strategy

Created by Interaction Routing Designer (IRD). Uses strategy-building objects (routing rules, interaction data, business rules, and so on). Can be contained in an interaction workflow. Can apply logic (segmentation, conditional branching). Can deliver to an agent or other target. Used within an interaction workflow, can switch between business processes.

screening rule

Pattern matching for character strings in e-mails.

segmentation

Function and object of routing strategy, applying conditional branching to the routing of an interaction (if it has attribute A, do X; if B, do Y; otherwise do Z).

snapshot

A list of all the interactions in Interaction Server's database that meet specified conditions at a given time. The agent application requests a snapshot from Interaction Server and uses the results to populate the list of interactions that display on the Agent or Supervisor desktop. See also the "Snapshot Operations" section of the "Basic Interaction Models" chapter of the *Genesys 7 Events and Models Reference Manual.*

standard response

Item in the Standard Response Library, which stores prewritten responses for use as suggestion to agent, acknowledgment, and/or autoresponse. Each standard response is assigned to exactly one category in the system (however, a category may have zero or many standard responses assigned to it).

suggestion to agent

A possible use for standard responses: text displayed in an agent's desktop application as a suggestion for use in replying to an interaction. The agent can paste the text into an e-mail message or a chat reply, or read from it during a voice interaction.

terminal category

A category that contains no subcategories; a leaf on the category tree. A category that contains subcategories is a *nonterminal* category.

third party

It is important to distinguish Third Party *Components* from third-party *applications*.

- Multimedia Third Party Components is a collection of files, mostly Java libraries, that Multimedia uses for its normal operation. Third Party Components is installed as part of the integrated install, as described in the *Multimedia 7.6 Deployment Guide*. The Third Party Components installation package places these files in various locations on the host machine.
- `ThirdPartyApplications` is a directory on the Multimedia product CD that contains the web server Apache, the application container Tomcat, and various supporting files. Multimedia requires a web server and an application container, but Apache and Tomcat are not the only supported types, and you are not required to install the particular copies in this directory; they are provided as a convenience.

training

(Genesys Content Analyzer) Process that the Content Analyzer follows to create and refine models: it works its way through a training object, which is a number of e-mails that have been classified according to a category tree. These may be actual e-mails that agents have classified, or they may be created especially to be used in training.

training object

A category tree plus a group of text objects that are classified according to the category structure. Typically the text objects are e-mails, but you can also include standard responses in the training object. Training operates on a training object to produce or refine a classification model, which can then be used to classify new e-mails.

view

A configurable node in an interaction workflow. Each queue defined in an interaction workflow is associated with one or more views. A view defines the queue-processing rules: what interactions should be selected from the queue and in what order.

VRP

Virtual Routing Point, a type of DN. A virtual device (not a device in a switch), not associated with any particular target, where customer interactions wait while Universal Routing Server makes routing decisions.

web collaboration

The ability for agents and customers to simultaneously navigate shared web pages; sometimes also known as co-browsing or conavigation. In release 7.6 this functionality is provided by two components, Genesys Co-Browsing Server and KANA Response Live Server. These components are provided as Genesys Web Collaboration, which is an option that you can add to either Genesys Web Media or Genesys Inbound Voice. See also "Multimedia and the CIM Platform" on page 10.

workbin

Contact center object holding interactions for later processing by a particular agent (or agent group, place, or place group). Agents can use workbins to store interactions that they have started working on and wish to continue working on at a later time. Interactions can also be distributed to workbins by Universal Routing Server. A workbin is like a queue in that it holds interactions. A workbin differs from a queue as follows:

- It is associated with a particular agent/place/group and its major function is to hold interactions for that agent/place/group to process.
- Agents can view the entire content of the workbin and pull interactions from it in any order. Agents can also pull interactions from queues, but only in the order defined by the queue.

workflow strategy

See interaction workflow.

Index

Symbols

:. 54
. 54
<$ 52
$> 52

A

accuracy
  of model 85, 110, 119, 241
  See also confidence, precision, recall
acknowledgement 241
Acknowledgment
  IRD object 46
Active
  standard response attribute 36
Add outbound categorized messages (check box) 87
Add uncategorized messages for manual clustering (check box) 87
age()
  translation function 219
agent
  field code object 46
agent capacity 241
answer
  in FAQ object 36, 145, 150
Application Cluster 163
Approved
  standard response attribute 36
archiving 242
attached data
  in inbound interactions 186
  in Interaction Server database 189
  limitations 180
AttachedData property 47
audience
  defining 12
AutobotAgentLoginName 46
Autoresponse
  IRD object 46, 56
autoresponse 242
Average Results
  Knowledge Manager tab 130

B

backup configuration 167
boolean IgnoreCase 71
brackets
  angle 14
Business Attribute 214–216
business process 242

C

cache (Interaction Server database) 242
category 19–40, 84, 242
  converting from ICS 6.5.x 234
  importing and exporting 77–81
  nonterminal 246
  terminal 246
Category Confusion
  Knowledge Manager tab 132
chapter summaries
  defining 12
character sets 171
Chat Server
  high availability 168
CIM (Customer Interaction Management) Platform 10
Classification Server 20, 84, 243
  administration 174
  high availability 168
Classify
  IRD object 140
co-browsing 11, 242
Co-Browsing Server
  high availability 168
  limitations 181

Condition tab for view 217
confidence 243
  rating in FAQ object 145
  rating of model. 131
  See also precision, recall
contact
  creation 203
  field code object 45
  identification 197–202
  matching attributes for chat session. . . . 178
content analysis 243
  Japanese 117
copying
  training objects 88
Correct in Top N
  Knowledge Manager tab 134
cross-validation. 119
current_time
  translation function 217
custom variable 46, 55
  restrictions on naming 46
Customer Data in Standard Responses . . . 46

D

database
  performance 170
  transition tool 223
  Universal Contact Server 20
DB2. 177
delimiters, field codes 52
delivery status notification 187
dictionary 30
document
  conventions 13
  version number 13
document errors
  commenting on 14
DSN (delivery status notification) 187

E

E-mail Server Java
  enable-same-mail-from-mailboxes option . . 61
  handling unparsable e-mails 182
  high availability 168
  limitations 176
empty
  translation function 220
Enabled
  screening rule attribute 36, 61
ESP 166, 192, 195, 243
Exact address match 61
expiration date
  standard response attribute 36
External Services Protocol
  See ESP

F

FAQ
  attribute of category 27
  field codes inapplicable 47
  objects. 77, 142–156
  standard response usage type 36
field code. 30, 44, 243
  Editor 48
  importing and exporting 77–81
firewall 175
format operator. 54
formula 53
frequency
  rating in FAQ object 145
function
  in field codes 54
  in screening rules 70

G

Genesys Content Analyzer 83–157, 243

H

high availability. 168

I

IgnoreCase. 71
importing
  Preserve uniqueness checkbox. 80
  Update screening rules checkbox. 81
InboundDisposition. 187
InboundReport 187
inheritance
  of categories 24
interaction
  field code object. 45
Interaction Routing Designer. 244
Interaction Server
  administration 174
  Application type 160, 174
  balancing DB Servers 167
  custom database fields 188
  high availability 168
  limitations 180
  utility mode 190
Interaction Server Proxy
  high availability 168
interaction workflow 244

J

Japanese language
  Content Analyzer 117

K

Key 71
Knowledge Manager 19–141
  limitations 177
  Tools menu 22

L

language
  Business Attribute 30, 216
  classifying by 115
  detection model 141
  in Knowledge Manager 23, 85
  in spelling checker 30
lexical analyzer 116

M

Mail Editor 92
MDN (message disposition notification) . . . 187
media
  channels 10
  customizing contact identification/creation 201
message disposition notification 187
Microsoft SQL 170
model
  rating 119
models
  accuracy ratings 128
  defined 244
  importing and exporting 77–81
  testing 120

N

naming restrictions
  custom variables 46
  impact on importing/exporting 81
  Knowledge Manager objects 26, 234
natural language processing 245
NDR (non-delivery report) 185
nonterminal category 23, 246
not_empty
  translation function 221

O

online 245
operational data 245
operator 53
  in screening rule 71
  precedence in field codes 53
  precedence in screening rules 72
Order
  screening rule attribute 61

P

precision 131, 245
pruning 242

Q

question
  in FAQ object 27, 145, 150
queue 245

R

recall 131, 245
regular expression 73, 75
rendering 45, 49, 52, 246
reporting on model ratings 135
Results by Category
  Knowledge Manager tab 133
routing strategy 246

S

Screen
  IRD object 61
screening rule 20, 58–76, 246
  attaching to a category 62
  converting from ICS 6.5.x 233
  importing and exporting 77–81
  Screen mailbox 61
screenshots 22
security 175
segmentation 246
Short Message Service (SMS) 42
snapshot 246
spelling checker 30
standard response 28–39, 246
  converting from ICS 6.5.x 234
  finding 40
  HTML format 28, 32
  importing and exporting 77–81
Standard Response Editor 29
stop words 117
Subject
  in standard response 30
suggestion to agent 36, 246
switch 160
symbols
  <> 14

T

Tenants tab 159
terminal category 23, 246
third party 247
thread 244

time_in_queue
  translation function 219
time_in_same_queue
  translation function 220
timestamp
  translation function 217
timestampadd
  translation function 218
timestampdiff
  translation function 218
TO Data Analyzer 95
Tools menu
  Knowledge Manager 22
training 247
  quality 110
training objects 84–115, 247
  copying 88
  importing and exporting 77–81
  large 118
  merging 122
  messages included 88
Training Server 20, 84
  high availability 168
translation function 216
Translations 217
typographical styles 13

U

Unicode 181
Universal Contact Server (UCS)
  data in standard responses 45
  database transition tool 223
  high availability 168
  importing Knowledge Manager objects . . . 80
  limitations 160, 176
Universal Contact Server Proxy
  high availability 168
UpdateBusinessData
  IRD function 205
UpdateUserData
  IRD function 205

V

version numbering
  document 13
view
  business process object 205, 207, 210, 213, 214, 247
  Conditions tab 217
VRP 247

W

Web API Server
  high availability 168
web collaboration 11, 15, 248
  See also co-browsing
workbin 248