



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

eServices Manager Plug-in for GAX

eServices Digital Administration 8.5.0

3/10/2022

Table of Contents

eServices Manager Plug-in for Administrator	3
Standard Responses	6
Plan and Build a Category Tree	7
Create a Standard Response	12
Set the Standard Response Properties	15
Create the HTML Version	17
Add Attachments	19
Create and Manage Multiple Versions	20
Standard Responses for SMS Gateways	21
Field Codes	23
How to Create and Insert Field Codes	25
Field Code Variables	28
Using Formulas in Field Codes	31
Field Code Examples	35
Screening Rules	38
How to Create and Test Screening Rules	39
Screening Rule Reference	48
Functions, Arguments, and Operators	50
Regular Expressions	53
Examples of Screening Rules	55

eServices Manager Plug-in for Administrator

The eServices Manager Plug-in gives you the tools to:

- Respond to incoming interactions using pre-written **Standard Responses**.
- Customize the Standard Responses using **Field Codes** to add a personal touch.
- Create **Screening Rules** in order to screen interactions for specific words or phrases, which you can then use to decide how to handle the interaction.

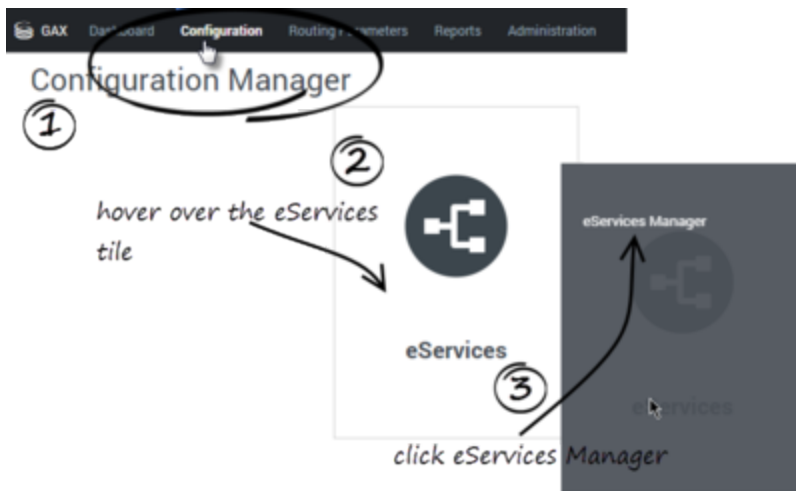
Important

- The eServices Manager Plug-in for Administrator allows users to create and store Knowledge Manager objects (categories, screening rules, standard responses etc.) in a multi-tiered hierarchical structure. Composer and Orchestration Server have not yet been upgraded to accommodate this new hierarchical structure. For this reason, customers using Composer and Orchestration Server are advised NOT to design their screening rules in multilevel hierarchy trees. Instead, keep them all at the top level in the Category Tree.
- This plug-in is the first phase of a migration away from the existing Knowledge Manager application towards a browser-based user experience. This initial release of the eServices Manager Plug-in for Administrator offers essential Knowledge Manager functionality; additional functions will be added in subsequent releases.
- Genesys Content Analyzer is not included in this release of the eServices Manager Plug-in for Administrator. You can find it in the legacy Knowledge Manager application. Genesys Content Analyzer uses natural language processing to analyze incoming interactions and assign them to categories in a category tree. For details about this functionality, see *Genesys Knowledge Manager User's Guide*.

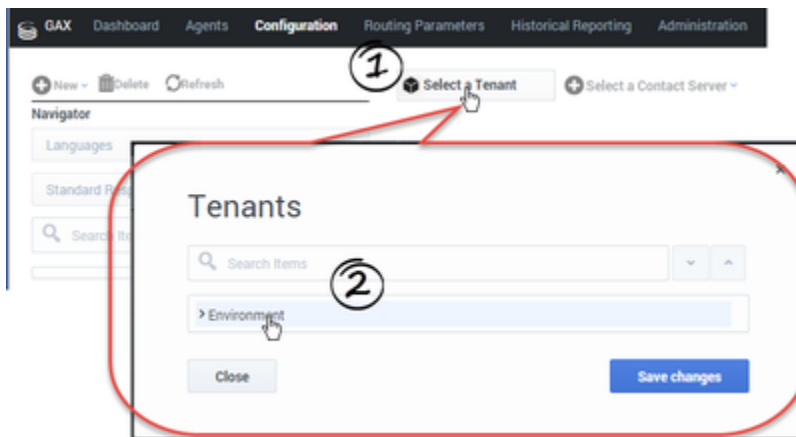
Let's Get Started

If the eServices Manager Plug-in isn't yet installed, see **Managing Plug-ins** for instructions.

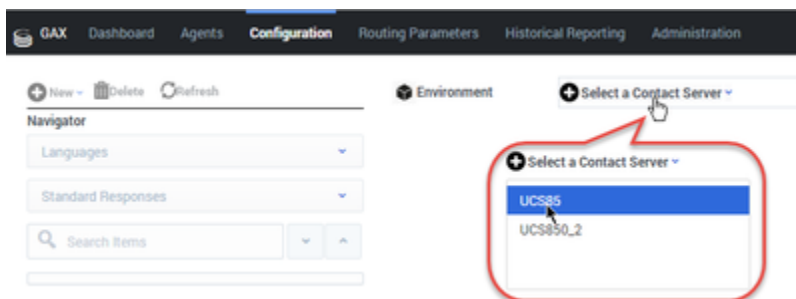
Then start Genesys Administrator Extension and open the eServices Manager Plug-in:



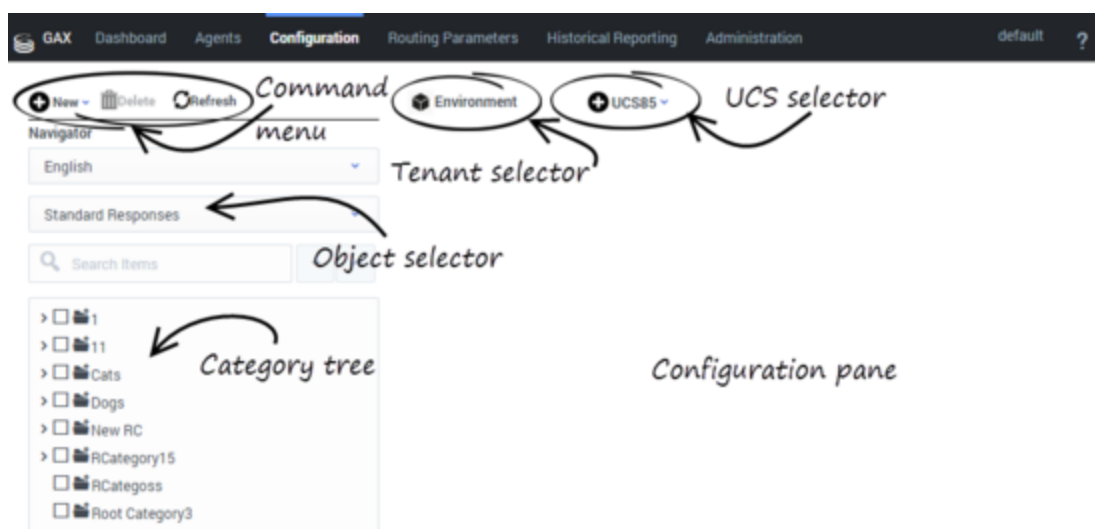
Now, select your Tenant...



and your Contact Server:



The eServices Manager window includes the following areas:



What Next?

- **Working with Standard Responses** explains how to create and edit Standard Responses and how to create the category tree structure you will use to organize your Standard Responses, Field Codes, and Screening Rules.
- How to create and edit **Field Codes**, which enable you to insert personal information, such as names, into your Standard Responses.
- **Working with Screening Rules** explains how to create and edit Screening Rules, which enable you to analyse what incoming messages are about and to handle them accordingly.

Standard Responses

Standard Responses enable you to send welcoming, helpful answers to frequently-submitted queries from customers.

The high-level process

To create Standard Responses you do the following high-level steps:

1. **Plan and build your Category Tree structure.** A Category Tree provides the framework for organizing your Standard Responses.
2. **Create your Standard Responses.**
3. **Create Field Codes** and then add them to the Standard Responses. Field Codes enable you to personalize your Standard Responses.

Plan and Build a Category Tree

A Category Tree consists of one or more *root categories*, each of which can have subcategories under it. Standard Responses are nodes under categories or subcategories in the Category Tree.

Step 1: Planning

We'll start by planning your Category Tree. To make your Standard Responses useful, you need root categories and subcategories that make sense in your business.

For example, you might create root categories for business units such as Sales, Service, and Billing. Then create useful subcategories. You might want to separate out customer type, such as Platinum, Gold, and Silver customers; or maybe it would work better for you to separate types of products, such as Pet Food, Pet Toys, and Cleaning Supplies; or you might want to respond differently to customers in different locations.

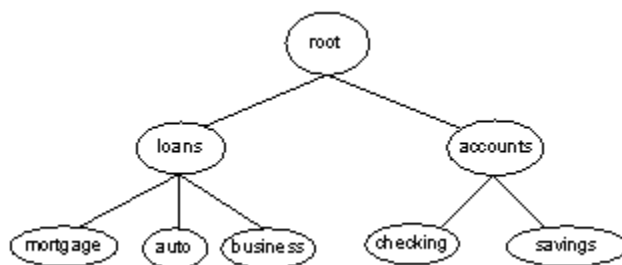
- Make sure that all the Standard Responses you need will fit within the categories you create.

Tip

Plan your category structure ahead of time. You can edit it, but a little thought now will save time a frustration later.

To read more about how category structures work, see **[+] How Categories Structures Work**

In general terms, a *category* is a unit of knowledge. Categories are organized in a tree structure; "Example Category Tree" shows an example.



Example Category Tree

Genesys eServices uses category trees to organize and provide 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.

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 "Example Category Tree", **savings** is a child of **accounts**, and accounts has the two children **checking** and **savings**.

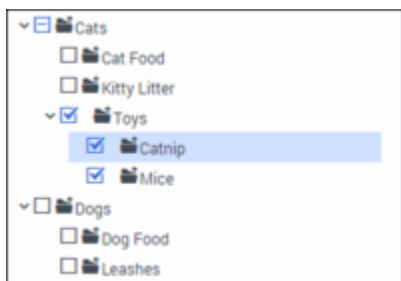
A category tree is specific to a tenant and a language. Each tenant/language pair can have multiple category trees.

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.

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.

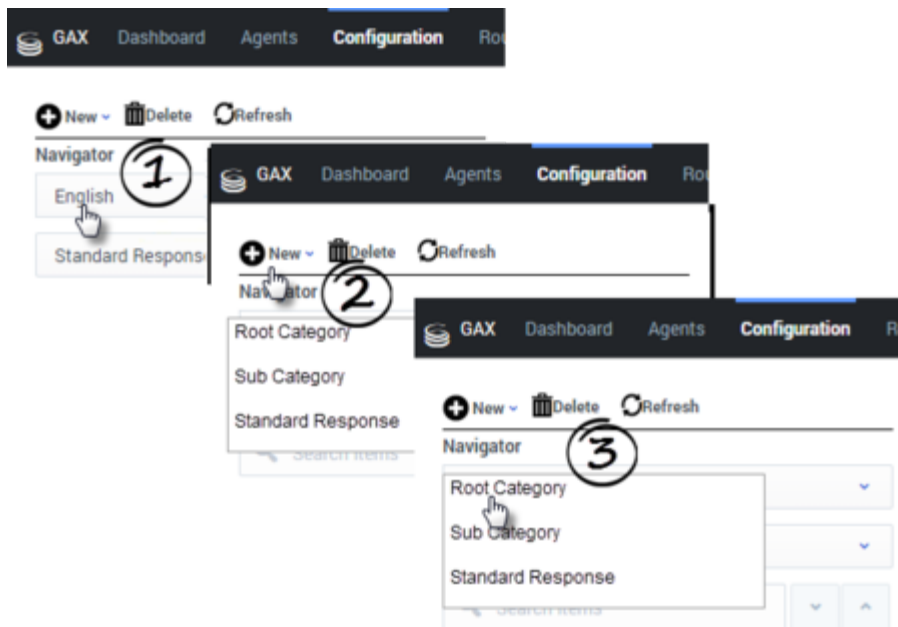
Step 2: Build your Category Tree

You've done your planning. Now let's make it happen. For our example, we are going to create two new root categories, Cats and Dogs, with subcategories, and sub-subcategories:



First we'll create our two root categories:

1. Make sure you have selected the right Tenant and Contact Server.
2. Select the language. Note that in this release, the language choice does not affect how messages are handled.
3. Select **New > Root Category**.



4. Then enter the root category name, Cat, and click **Save**. Repeat the same process to create the Dog root category.

The screenshot shows the 'New Root Category' form. It has a 'Name' input field and a 'Save' button. The 'Environment' is set to 'UCS85'.

5. Now create the subcategories. For each subcategory, highlight the correct root category and then follow the same process as above, but select **New > Sub Category** instead of **New > Root Category**.

Characters Allowed in Object Names

Important

Names of categories, like those of all eServices Manager objects, can consist only of alphanumeric characters (A-Z, a-z, 0-9), plus the characters shown in "Characters Allowed in Object Names".

Name	Character	Name	Character
Hyphen	-	Exclamation point	!
Number sign, pound	#	Dollar sign	\$
Caret	^	Asterisk	*

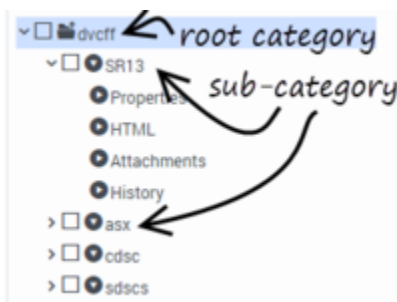
Name	Character	Name	Character
Underscore	_	Curly brackets	{ }
Angle brackets	< >	Period, full stop	.

Names can be no more than 64 characters long.

- To change a category name, select it, edit the name in the text box, and click **Save**.
- To delete a category, select it and then click **Delete**.
 - Be careful to select only the category check boxes you want to delete!
 - Deleting an upper-level category also deletes all the categories under it.

Important

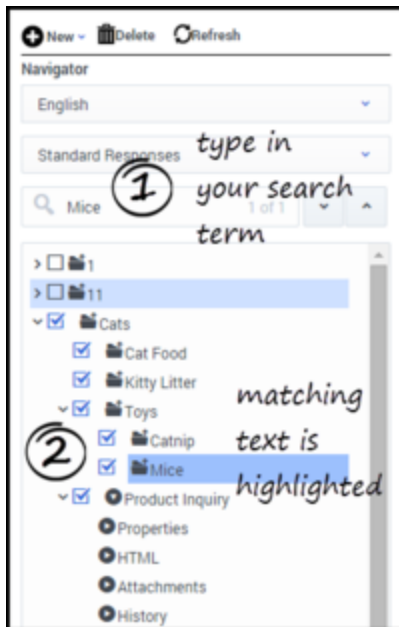
To open the root category and show the categories beneath it, click a caret mark (>).



Search the Category Tree

To locate a category, Standard Response, Field Code, or Screening Rule:

- Select the type of object you want to search for and then enter your search term.



Next Steps

- [Creating Standard Responses](#)
- [Standard Responses for SMS Gateways](#)
- [Personalizing Standard Responses with Field Codes](#)

Create a Standard Response

A Standard Response is an item in the Standard Response Library, which stores prewritten responses for use as suggestions to agents, acknowledgments, and/or autoresponses. 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.

You can use Standard Responses for any of the eServices channels: eMail, Chat, Social Engagement, or SMS.

It takes just a few steps to create a Standard Response. The Process Overview below lists the main steps. We'll take each one in sequence, or you can skip to the information you need right now.

Process Overview

- [Create a Standard Response](#)
- [Configure the properties for your Standard Response](#)
- [Enter the HTML version](#)
- [Add attachments](#)
- [Create additional versions and retrieve prior versions](#)
- [Create and add Field Codes](#)

To create a Standard Response:

1. Select the correct Category Tree node, and then select **New > Standard Response**. The first Standard Response configuration page appears on the configuration pane.

2. Enter the basic information for your Standard Response on the configuration pane.

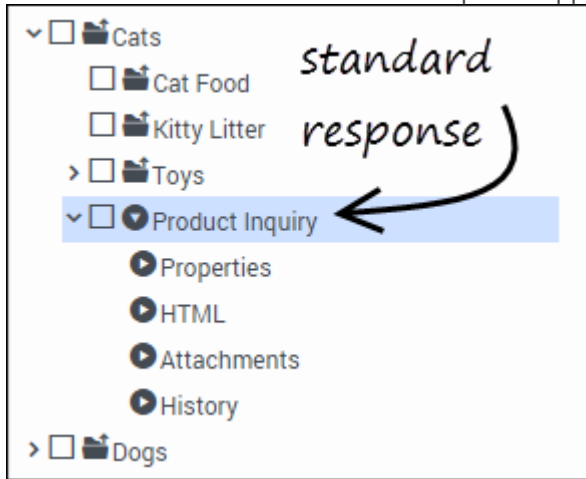
The text box is for the plain text version of your Standard Response. After you save these basic parameters for your Standard Response, you will be able to create the HTML version of your response and add other important information.

Important

The Standard Response name can consist only of alphanumeric characters (a-z, A-Z, 0-9), hyphen, underscore, and space. For more information about characters you can use in eServices Manager objects, see [Characters Allowed in Object Names](#).

You can also specify a Subject line in any IRD strategy object that has a **Format** tab (see the [Universal Routing 8.1 Reference Manual](#)). If you do, this overrides any Subject line that is specified for the Standard Response in eServices Manager.

3. Click **Save** and the new Standard Response appears in the Category Tree.



4. To edit the standard response, select it, make your changes, and then click **Save**.

Plain Text and HTML Versions

The basic information includes the plain text version of the response. You will have the option of creating an **HTML version** after you create and save the plain text version.

When E-mail Server 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.

Important

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 eServices Manager.

Field Codes

You will probably want to use **Field Codes** to personalize your Standard Responses. We'll be creating Field Codes after creating some Standard Responses, and then insert the Field Codes into them.

Set the Standard Response Properties

To configure the properties of your Standard Response:

- Click **Properties** under your Standard Response in the Category Tree to open the Properties configuration page.

Environment UCSB5

Properties for Standard Response

Usage	Active
<input checked="" type="checkbox"/> Acknowledgement select check boxes for how this response will be used	<input type="checkbox"/> type only ONE response can be Active; that is, it's the default response for that Usage type
<input checked="" type="checkbox"/> Autoreponse	
<input type="checkbox"/> Suggestions	
Owner: default the response creator's username	
Start Date: 26 May 2015 by default, today's date	
Expiration Date: 26 May 2015 to enter an expiration date, clear the Never Expire check box	
Never Expire: <input checked="" type="checkbox"/>	
Approved: <input checked="" type="checkbox"/> only Approved responses can be used in Routing Strategies	

Save

The possible Usages are:

- Acknowledgment—The standard response may be sent to acknowledge receipt of an incoming interaction.
- Autoreponse—The standard response may be used as an automatic response to an incoming interaction.
- Suggestions—The standard response may be offered to agents as suggested wording to use in their own replies to interactions.

Each category in the Category Tree may have multiple standard responses of each Usage type.

About the Active Parameter

For each Usage type, you must specify whether this standard response is the Active one. Only one standard response of a given Usage type can be Active. When the system needs to send a Standard Response of a specific Usage type automatically, it sends the one marked Active.

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, eServices Manager offers to take the previously Active Standard Response out of Active status.

About the Expiration Date

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 does not send the standard response, but returns an error message.

Next Steps

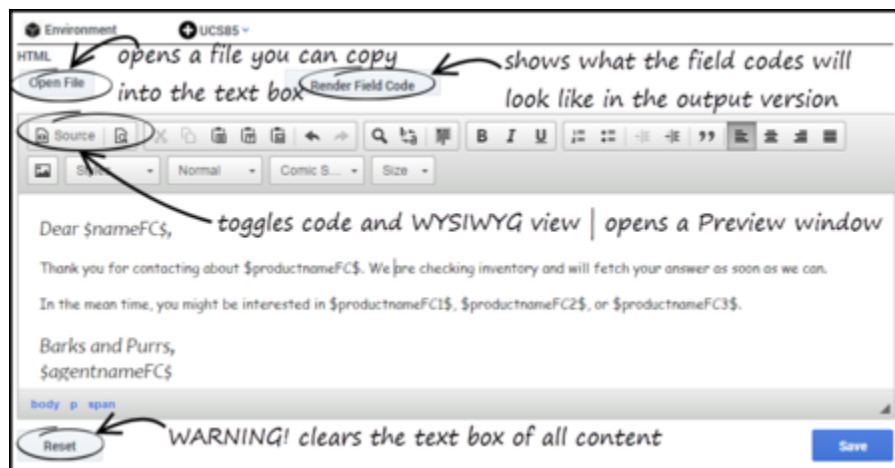
- Create an **HTML version** of the standard response.
- **Add an attachment** to the standard response.
- **Maintain multiple versions** of the standard response.
- Create **Field Codes** to use in your standard responses.

Create the HTML Version

To create the HTML version of your Standard Response, click **HTML** under your Standard Response in the Category Tree to open the HTML configuration page.

Most of the buttons provide commonly-used editing functionality, including the option to insert a link to an image. If you aren't sure what a button does, hover over it to open a tooltip.

The figure below explains buttons with more specialized functionality.



Add Text Content

You can


- Type or paste in any text as plain text.
- Paste formatted content from Microsoft Word.
- Type or paste HTML code into the code view.
- Import an existing HTML file by clicking **Open File**.

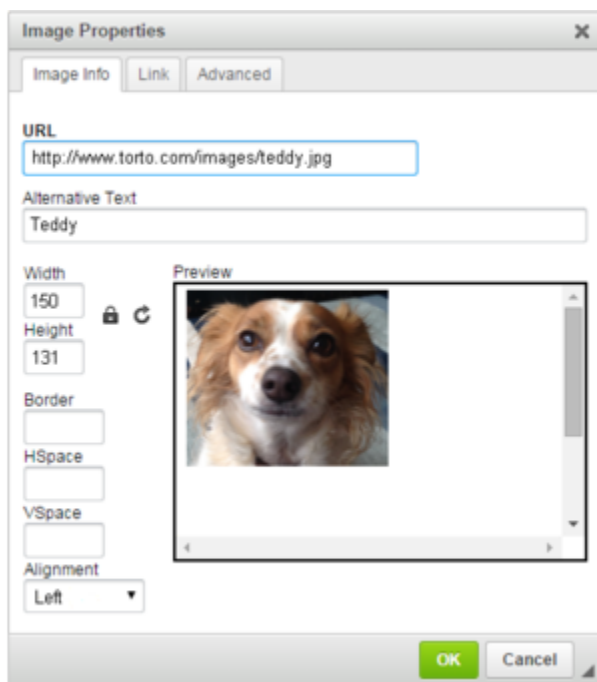
Important

- 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.

- Make sure that the content of the HTML version, including field codes, matches the plain text version that you created on the main configuration page tab. This is important because 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.

Add an Image

To add an image, click the  button, then configure the dialog box that opens:



Next Steps

- [Add an attachment](#)
- [Review the history and manage versions](#)
- [Create Field Codes](#)

Add Attachments

To add an attachment to your Standard Response:

Click **Attachments** under your Standard Response in the Category Tree to open the **Attachments** configuration page.



- To add an attachment, click **Attach**.
- To remove an attachment, select it, then click the trash can icon.
- To view an attachment, select it, then click **Open**. This opens the attachment for viewing only. You cannot save any changes that you make to the attachment.

Next Steps

- [Review the history and manage versions](#)
- [Create Field Codes](#)

Create and Manage Multiple Versions

To create multiple versions of your Standard Response:

Click **History** under your Standard Response in the Category Tree to open the History configuration page.



After you save multiple versions, use the **Remove** and **Restore** buttons to manage them.

- To choose the version that appears on the Standard Response pane, select the check box for the version you want and click **Restore**.

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

- Version 1 of a standard response has an expiration date of May 31, 2015.
- Version 2 with has **Never expire** selected (no expiration date).

Now, restore Version 1.

The restored Version 1 is set to **Never expire**. The former expiration date is not restored.

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 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 8.1 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 <https://www.achab.it/achab.cfm/en/archive-server-for-mdaemon/documentation>.
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.

eServices Manager enables you to create a wide range of Field Codes types, from simple Field Codes that function similarly to a Mail Merge-type word processor feature to complex Field Codes that include multiple objects, formulas, and constants (see [Using a Complex Field Code](#) for an example).

Once you create a Field Code, you can use it in multiple standard responses.

The interface for creating Field Codes is simple; creating really useful Field Codes requires a deeper understanding of how Field Codes can be constructed. [How to Create and Insert Field Codes](#) provides step-by-step instructions, with links to detailed reference information at the relevant places.

The reference information consists of the following topics:

- [Field Code Variables](#)
- [Using Formulas in Field Codes](#)
- [Field Code Examples](#)

Field Codes Overview

The main use of field codes is to particularize standard responses.

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 outgoing text object, such as an e-mail that E-mail Server generates when triggered to do so by a routing strategy object.

The most common type of such text object is a standard response (triggered by an Autoresponse or Acknowledgement object), but you can also insert field codes into other types, such as chat transcripts, SMS messages, 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
 - RenderMessageContent

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 a text 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 text object.

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

Important

Field codes can be used in outgoing text objects only.

A complete reference list of field codes is available in the [Genesys eServices Field Codes Reference Manual](#).

How to Create and Insert Field Codes

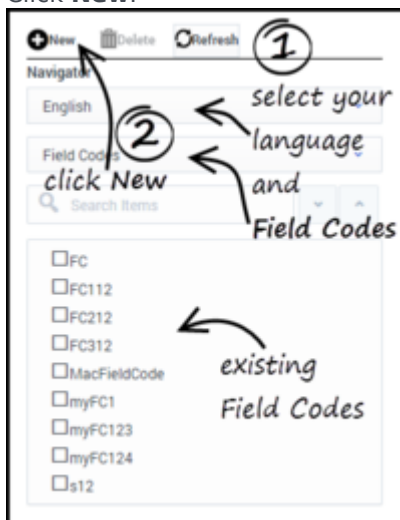
Using Field Codes includes:

- [Creating Field Codes](#)
- [Inserting Field Codes into a Standard Response](#)

Creating Field Codes

From the eServices Manager browser window:

1. Select the language and **Field Codes** from the drop-down menus.
2. Click **New**.



3. To configure your new Field Code, enter the properties you need in the Field Code configuration pane and then click **Save**.

The screenshot shows the 'Field Code' dialog box. It has fields for 'Name' (containing 'CustomerAccount') and 'Description' (containing 'customer account number'). Below these are two sections: 'System Variables' with a dropdown showing 'Contact.Id' and an 'Insert' button, and 'Custom Variables' with a dropdown showing 'CustAccount' and an 'Insert' button. At the bottom is a 'Text' field containing 'Agent.FirstName+CustAccount'. Annotations with circled numbers point to specific elements: (1) points to the 'Name' field with the text 'Choose the variables you want'; (2) points to the 'Insert' button in the 'Custom Variables' section with the text 'then click Insert'; (3) points to the 'Text' field with the text 'the selected variables appear in the Text field'. Other buttons include 'Create Custom Variable', 'Check', 'Cancel', and 'Save'.

The new Field Code appears in the Category Tree pane. You can edit a Field Code anytime by double-clicking it in the Category Tree.

4. Enter a name and (optionally) a description for the field code.
5. Select System and/or Custom variables from the drop-down lists.
6. Click **Insert** to populate the text box with the selected variables.
See a detailed discussion of [variables in eServices Manager](#).

If the custom variable you need is not in the list, you can **[+] create a custom variable**.

- a. Click **Create Custom Variable**. The Custom Variable dialog box appears.
- b. Enter a name (required) and description (optional).

Important


The name must consist only of alphanumeric characters or underscores.

- c. Select **String** or **Integer** for the type.
- d. Enter a default value. This is mandatory.
- e. Click **Add**. The new custom variable appears in the list.

Important

You can also use this dialog box to edit and delete (Remove) existing custom variables.

- f. Click **OK**.

- g. Refresh  the browser window if the new custom variable doesn't immediately appear in the drop-down list.

7. Enter any other desired text in the Text field. This text must conform to the rules described in [Using Formulas in Field Codes](#).
8. Click **Check** to verify that the field code is well-formed (that is, that it has no typographical errors, missing parentheses, and so on).

For detailed help constructing Field Codes, see [Using Formulas in Field Codes](#). For examples showing the use of a custom variable and of a complex Field Code, see [Field Code Examples](#)

Inserting Field Codes into a Standard Response

To insert Field Codes:

1. Select a Standard Response from the Category Tree or else [create a new Standard Response](#).

Important

Standard Responses that are intended for use in FAQ objects should not contain field codes.

1. Click **Insert Field Code** to display a list of all the available field codes.
2. Select a field code and click **OK** to insert it, together with its required delimiters (<\$ \$>), into the Standard Response.
3. Click **Check Field Codes** to see the standard response with the Field Codes rendered, showing the default values for each Field Code.

Field Code Variables

Using Field Code Variables includes:

- [Using UCS Data as System Variables in Standard Responses](#)
- [Custom Variables](#)
- [Using Your Own Data in Standard Responses](#)

Using UCS Data as System Variables in Standard Responses

In the example given in [Field Codes Overview](#), the `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, such as 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

Important

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 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 eServices 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 [Using a Custom Variable](#). For a complete description of the Routing objects that can use custom variables, see the [Universal Routing 8.1 Reference Manual](#).

Important

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 eServices 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:

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

Using Formulas in Field Codes

In addition to system variables such as `Contact.FirstName`, field codes may contain formulas. This section provides an outline of formula usage. Details on many of these topics are provided in the [Genesys eServices Field Codes 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 eServices 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

To summarize field code syntax:

- 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 available.

HTML in Field Codes

With special configuration, field codes can contain HTML markup; for example, you could have a field code `<$my.agent.signature$>` defined as

```
Sam Agent<BR />
Acme Products<BR />
29 Exterior Blvd<BR />
Springfield, CX 09090<BR />
```

To enable this, you must use the Java property `-Dsr1-field-code-allow-html=true`, in one of the following ways:

- Add it to the `JavaArgs` section of `ContactServerDriver.ini`
- Add it as an argument to the startup command line in `contactServer.sh`.

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 the [Genesys eServices Field Codes 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 the [Genesys eServices Field Codes 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 "[Functions](#)" in the [Genesys eServices Field Codes Reference Manual](#).

Using Objects

All object/property pairs are also available in the Variables drop-down menu in the eServices 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 the [Genesys eServices Field Codes Reference Manual](#).

Field Code Examples

This section presents examples of the use of field codes.

Using a Custom Variable

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

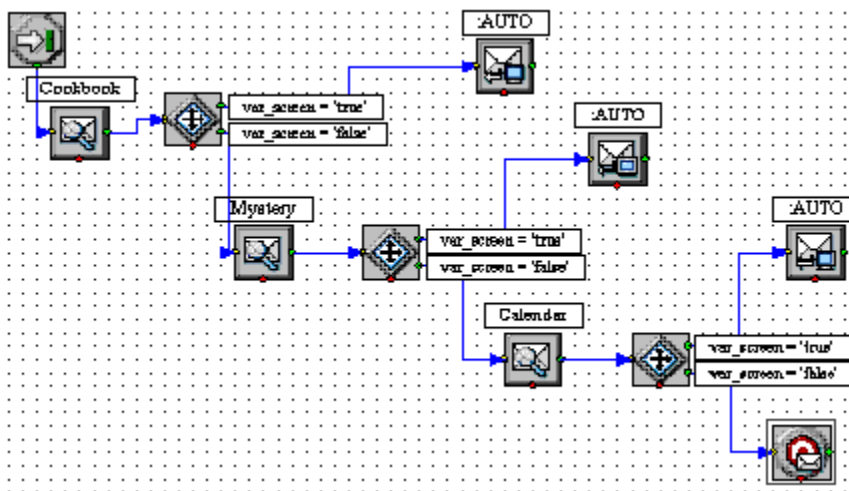
Prerequisites

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

1. In eServices Manager:
 - a. Create a custom variable called QueryTopic (see See Creating field codes).
 - 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:
 - e. Cookbook: `RegexFind("cook") || RegexFind("recipe") || RegexFind("food") || RegexFind("cuisine")`
 - f. Mystery: `RegexFind("murder") || RegexFind("crime") || RegexFind("case of the") || RegexFind("detective")`
 - g. Reference: `RegexFind("dictionar") || RegexFind("encyclopedia") || RegexFind("almanac")`
 8. 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.

The figure "Strategy Using Custom Variable" 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 ____*.



Using a Complex Field Code

The following is an example of a 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.

Next Step

- Go on to create and manage [Screening Rules](#).

Screening Rules

Screening rules scan an interaction and try to match either a destination address (who the message is going to, whether that is identified by an email address, a cell phone number, or some other parameter), a regular expression, or both. Screening is performed by Classification Server when it is triggered by a Screen Interaction object in a routing strategy.

A screening rule can optionally be associated with a category.

Important

Screening 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 which are screened will be e-mail messages; therefore, the terms *e-mail* and *message* are used interchangeably here, to refer to these interactions. In fact, whatever is said here about e-mail applies to any interaction that has associated text.

Screening Rules topics include:

- [How to Create and Test Screening Rules](#)
- [Screening Rules Reference](#)—This section includes:
 - What they check: [What Screening Rules Check](#)
 - The functions, arguments, and operators used in them: [Functions and Arguments](#)
 - A pattern builder to help formulate them: [Pattern Builder](#)
 - Regular expressions to use in them: [Regular Expressions](#)
- [Examples of Screening Rules](#)

How to Create and Test Screening Rules

The step by step procedure for creating a Screening Rule is quite simple. The power of Screening Rules lies in the many ways you can configure them. We'll start with the high-level procedure, then drill down as needed into more detailed explanations of the various parameters.

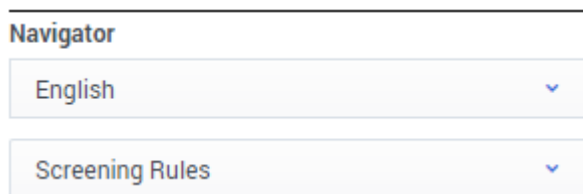
This page contains the following sets of instructions:

- [The High Level Steps](#)
- [Configuring a Screening Rule](#)
- [Assign Categories and Remove Category Assignments](#)
- [Test a Screening Rule](#)
- [Finding an Existing Screening Rule](#)

The High Level Steps

Note that the order of these steps matters!

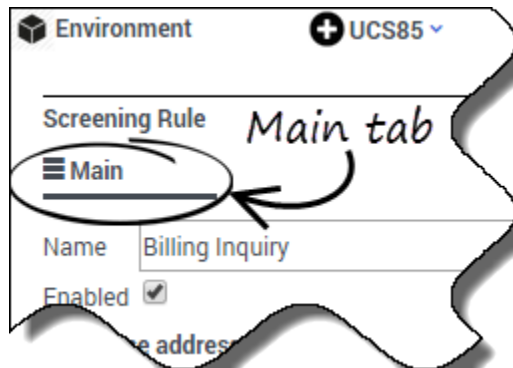
1. Start on the eServices Manager browser window. Select the correct:
 - Tenant
 - UCS Server
 - Language
2. On the **Navigator** drop-down menu, select **Screening Rules**.



3. On the Category Tree, check the boxes for the categories this Screening Rule should apply to.
4. Click **New**.
5. Select **Screening Rule**.
6. After configuring the Screening Rule—which will be explained shortly—click **Save**. The new Screening Rule appears in the Category Tree under the selected Root Category.
7. To edit a Screening Rule, highlight it in the Category Tree and then make the desired changes. Don't forget to click **Save** when you are done.

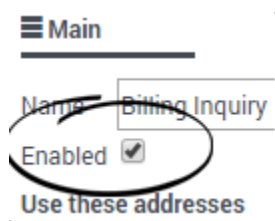
Configuring a Screening Rule

Configuring a Screening Rule is mostly done on the Main tab.



When you name your Rule, remember that you can only use **permitted characters in object names**.

Check the **Enabled** check box to make this Rule available when you add a Screen object to a routing strategy.



Once a strategy includes a Screen object that uses a particular Screening Rule, the strategy will continue to use the Rule whether it is disabled or enabled.

Do you need to use the Order text box?

if the Screening Rule is used in a Multiscreen routing object in which **All rules** is selected, use the **Order** text box to specify where in the sequence of rules this particular one should be applied.



In other cases, the **Order** text box is unused.

Do you need to screen the addresses the email is sent to?

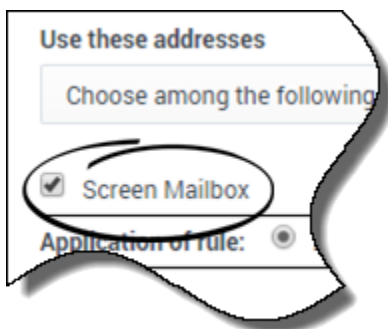
- If so, in the **Use these addresses** area, select an address from the left-hand window, then click **Add** to copy it to the right-hand window. You can add as many addresses as you need. If the address you need isn't there yet, you can define additional addresses in this configuration database object: **Business Attributes > EmailAccounts > Attribute Values > Annex > general > address**. or, instead of moving addresses from the left-hand window to the right-hand, you can directly type an address in the right-hand window.



- If you don't need to screen addresses, leave this section empty.

Does it matter if the address is exact?

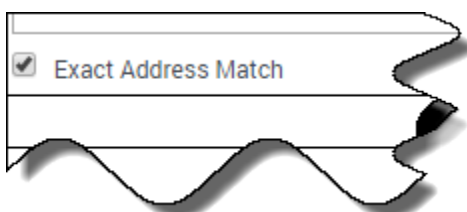
- If so, select the **Exact address match** box. The screening rule looks for messages having that exact address as a destination. For example, **xyz@domainname.com** matches **xyz@domainname.com** but not **abc.xyz@domainname.com**. This match is not case sensitive.



- If not, clear the **Exact address match** box. Then 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**.

Do you want to screen for the exact POP mailbox the email is sent to?

- If so, select **Screen mailbox** to make the rule match the POP box from which the e-mail entered the eServices 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.



Important

For this feature to work as expected, the E-mail Server **enable-same-mail-from-mailboxes** option must be set to `true`. With this setting, E-mail Server 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).

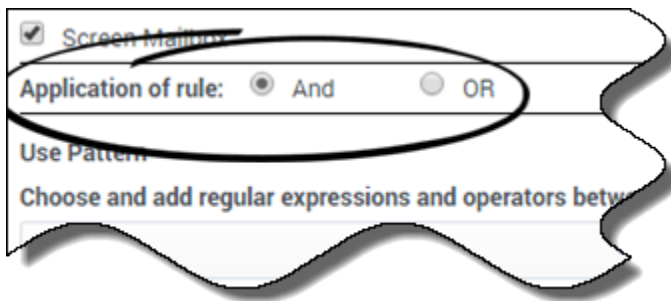
How to use the AND and OR radio buttons

- **AND**

Choosing **AND** makes eServices Manager use *both* the addresses selected in **Use these addresses** and match the pattern defined in **Use Pattern**.

- **OR**

Choosing **OR** makes eServices Manager use *either* the addresses selected in **Use these addresses** or match the pattern defined in **Use Pattern**.

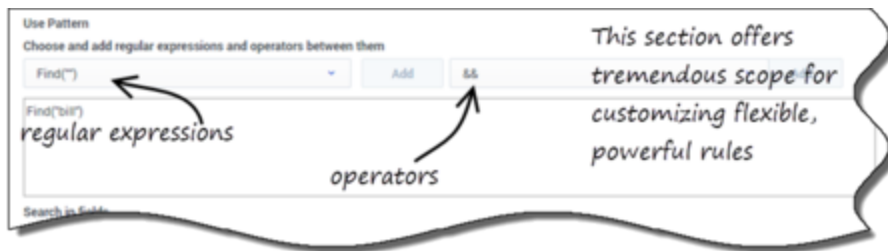


How to configure the Use Pattern area

This is the section where you can get really specific about what the Screening Rule should match. The Plug-in 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 below.



The drop-down lists in the **Use Pattern** area contain regular expressions (functions) and operators you can select to create your Rule. Select the regular expression you want, and then click **Add**. The expression appears in the text box. Do the same to add operators. You can also enter text manually.



After you select an expression 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.

Important

See [Screening Rules Reference](#) for an in-depth explanation of how to construct rules.

What message sections should the Screening Rule apply to?

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.



If you check multiple boxes, the Screening Rule can behave in one of two ways. See [Email Sections to Screen](#) for an explanation.

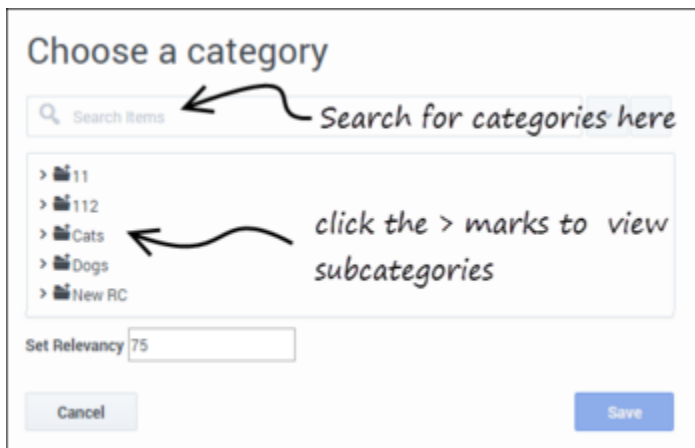
Assign Categories and Remove Category Assignments

You must select a category from the Category Tree to create a Screening Rule. However, you can change or add to the categories associated with the rule after you create it.

Click the **Categories** tab to open the **Choose a Category** dialog box.



When an interaction matches this screening rule, 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. *X percent* is the Relevancy value you set here.



Choose the desired categories and then click **OK**.

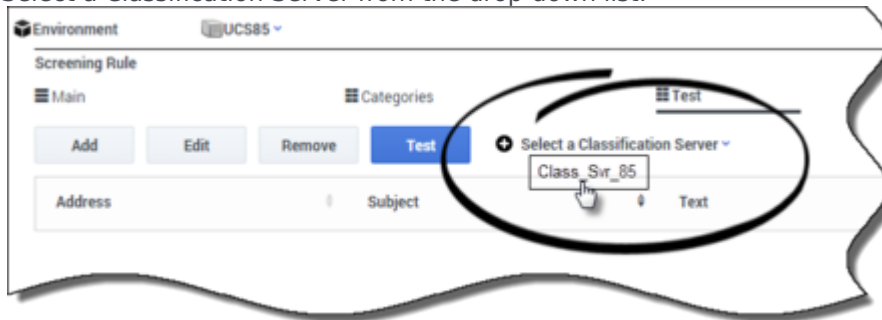
Test a Screening Rule

You will be testing your Screening Rule against some specially created text to see if it works as desired.

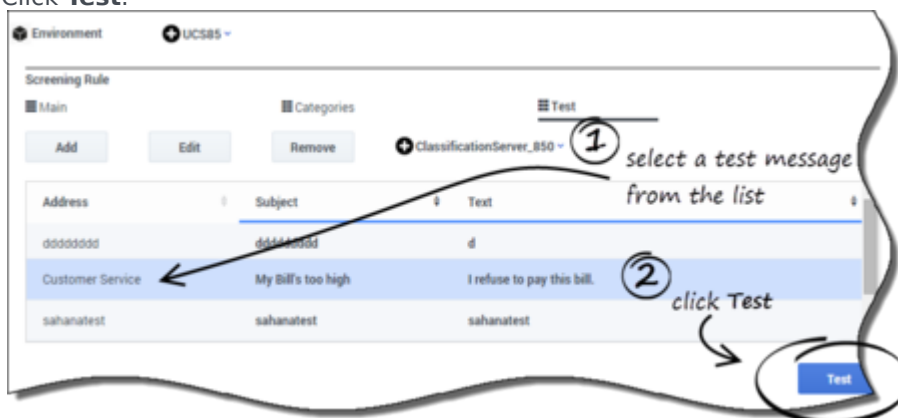
1. Click the **Test** tab to start testing.



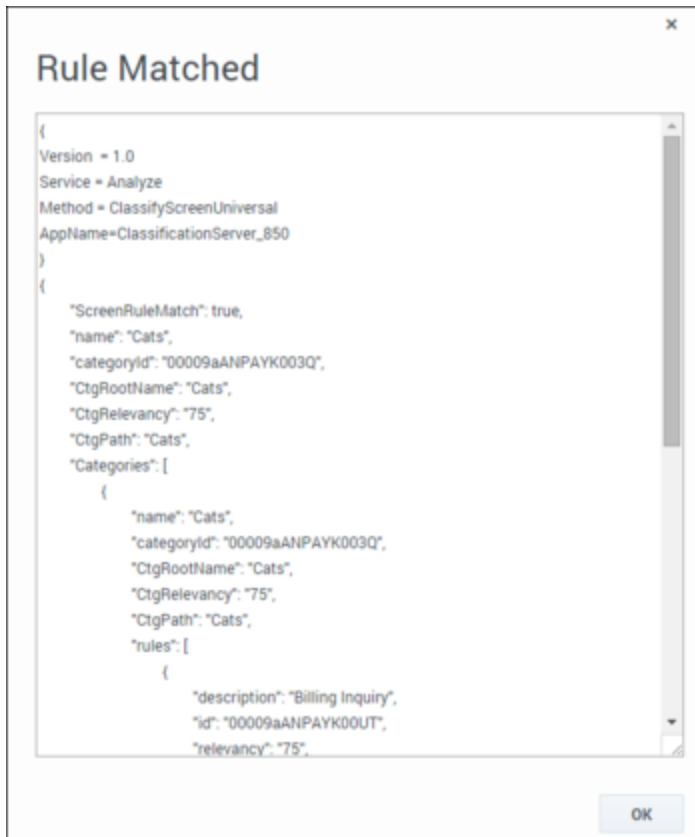
2. Select a Classification Server from the drop-down list.



3. Choose a test message from the list of those already created or click **Add** to create a new one.
4. Click **Test**.



If your test message fulfills the constraints you created for your rule, you will receive a detailed message indicating success.



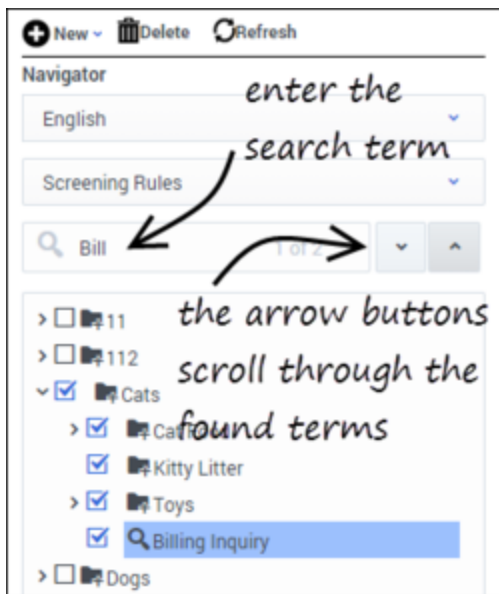
If you get a **Rule Not Matched** message, check the message for clues to the problem. If necessary, analyze your Screening Rule step by step to determine what needs to be edited to make it work the way you want.

Important

The test messages created by each user are stored separately, and are not synchronized between multiple eServices Manager instances. Test messages that you create are stored on your local machine under the default Documents folder of the logged-in Windows User folder, in the UserInfo file. This file is binary and must not be edited manually.

Finding an Existing Screening Rule

To search for a Screening Rule, make sure **Screening Rules** is selected in the **Navigator** drop-down list. Then start typing in the name of the rule in the **Search** field. The matching names are highlighted as you begin typing.



Next Steps

- Learn more about how screening rules work:
 - [Screening Rules Reference](#)
 - [Screening Rules Examples](#)
 -

Screening Rule Reference

This section provides reference information on the following topics relating to screening rules:

- What they check: [What Screening Rules Check](#)
- The functions, arguments, and operators used in them: [Functions and Arguments](#)
- Regular expressions to use in them: [Regular Expressions](#)

What Text Do 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" below, on how screening rules behave if two or more of the preceding are selected, and on the **Pattern is found in any selected field (OR relation)** checkbox.
- The destination address, if you have put anything in the **Addresses** area.
- 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 **Pattern** area. 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.

User data is first associated with the interaction by the media server when it creates that interaction. As an example, E-mail Server 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 Application object)
- To (MIME header field)
- Subject (truncated to 512 characters)
- FromAddress (personal part of From header field)
- FromPersonal (email 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) 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_FailureArgs

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

Email Sections to Screen

If you select more than one of the Subject, Body, and Header areas, the screening rule applies to all areas at once. There are two ways to implement this method:

- To enforce it for all screening rules, set the subject-body-header option for Classification Server to `true`.
- To enforce it for a particular screening rule:
 1. Leave **subject-body-header** set to `false`.
 2. Open the rule in the Screening Rule Editor.
 3. Select the **Pattern is found in any selected field (OR relation)** check box.

Important

Setting **subject-body-header** to `true` overrides any selection of the **Pattern is found in any selected field (OR relation)** check box for a particular rule.

Functions, Arguments, and Operators

- [Functions](#)
- [Arguments](#)
- [Operators](#)

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](#)). 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.

Important

`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](#)).
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."

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]`.

Important

In general usage, apart from Genesys eServices, regular expressions are case sensitive. However, in the eServices Manager Plug-in, regular expressions are not case sensitive unless you add `, false` as described in [IgnoreCase](#).

The table "Elements of Regular Expressions" lists some of the most commonly-used elements of regular expressions:

Elements of Regular Expressions

Symbol	Meaning	Example
.	Any character, including space	<code>b.t</code> matches <i>bat</i> , <i>bet</i> , <i>bit</i> , and <i>but</i> .
<code>\d</code>	Any digit	<code>\d\d</code> matches any pair of digits from 00 to 99.
<code>\s</code>	Space	<code>\d\s\d</code> matches 1 0, 5 9, and so on.
.	Zero or more instances of the preceding expression	<code>o*f</code> matches <i>oof</i> , <i>of</i> , and <i>f</i> . <code>me.*d</code> matches <i>med</i> , <i>mead</i> , and <i>meed</i> .
+	One or more instances of the preceding expression	<code>bre+d</code> matches <i>bred</i> , <i>breed</i> and <i>breed</i> .

Symbol	Meaning	Example
?	Zero or one instances of the preceding expression	<code>c?rude</code> matches <i>rude</i> and <i>crude</i> .
{x}	X instances of the preceding expression	<code>st.{2}k</code> matches <i>steak</i> , <i>stork</i> , and <i>stink</i> .
^	Any character except the following	<code>s[^e]t</code> matches <i>sat</i> , <i>sit</i> , and <i>sot</i> , but not <i>set</i> .
[]	Any characters or ranges within the brackets	Any characters: <code>b[aeiou]at</code> matches <i>boat</i> but not <i>brat</i> . Any range(s): <code>[0-9]th</code> matches <i>5th</i> , <i>6th</i> , <i>7th</i> . <code>[a-z]</code> matches any lowercase letter; <code>[A-Z]</code> matches any uppercase letter.
\	Turns off the special meaning of the following symbol	<code>*</code> matches the character <code>*</code> (asterisk); <code>\.</code> matches the character <code>.</code> (period or full stop).
	Or	<code>[b p]ig</code> matches <i>big</i> and <i>pig</i> . Do not be confused: <code> </code> means <i>or</i> in regular expressions, but <code> </code> means <i>or</i> 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 of Screening Rules](#).
- Regular expressions make use of many more special characters and operators than those listed in the table "Elements of Regular Expressions." Much documentation on regular expressions is available on the Web. Because Genesys eServices Manager uses Java classes for regular expressions, it is best to consult documents describing the particular version of regular expressions used in Java.

Examples of Screening Rules

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\
```

Important

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
```

The table "Phone Number Regular Expression" analyzes this regular expression.

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."