



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.

Genesys Knowledge Center User's Guide

Field Code Examples

12/19/2025

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.

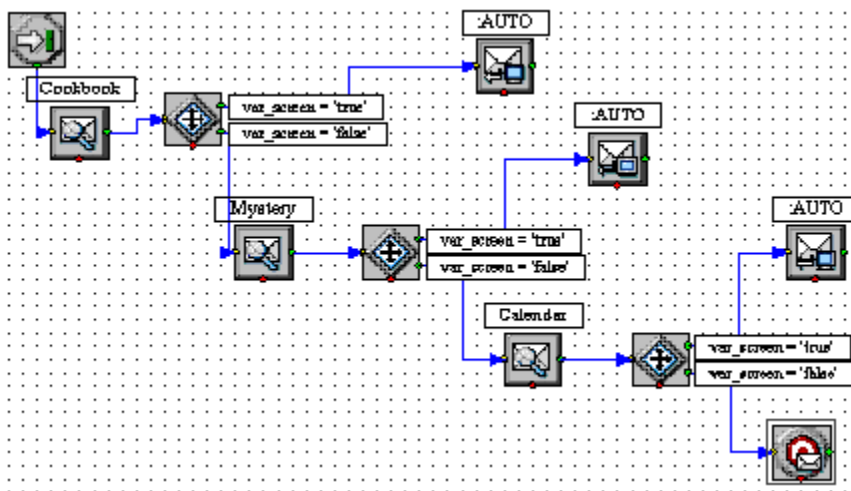
Prerequisites

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

1. In Knowledge Center CMS:
 - a. Create a variable called QueryTopic.
 - 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 email 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 emails, 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.