Genesys Knowledge Center User's Guide

Screening Rule Reference

4/16/2025

Contents

Screening Rule Reference

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" information 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, 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 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. Set "Find at least one of selected field"

> **Important**
> Setting subject-body-header to true overrides any selection of the "Find at least one of selected field" for a particular rule.

Functions, Arguments, and Operators

Functions

Screening rules can use three basic functions:

- Contain text (Find("<text>") in text mode), where <text> is a text string. It returns the result true if the interaction contains the exact string between quotes, ignoring case.
- Contain regex (RegExFind("<regular expression>") in text mode), 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.
- Matches Regex (RegExMatch("<regExp>") in text mode), 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**
>
> Contain regex and Matches Regex are the same except that Contain regex looks for a match anywhere in the body of the interaction, whereas Matches Regex 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)

Case insensitive (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.

The operators are as follows:

&& 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 regular expressions are case sensitive. However, in the Knolwedge Center CMS, regular expressions are not case sensitive unless you select this option as described in "Case insensitive".

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	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.

Symbol	Meaning	Example
^	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	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 of Screening Rules below.
- 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 Knowledge Center CMS 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

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	Type	Notes
\d\d\d	Three digits	n/a
\(\d\d\d\)	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	n/a
\d\d\d	Three digits	n/a
[\-\.]	Hyphen or period	Note again the need to use \
\d\d\d\d	Four digits	n/a

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."