eServices Manager Administrator's Guide

Screening Rule reference

2/4/2026

Contents

- 

- Administrator

View the functions, arguments, and regular expressions that you can use in screening rules, and how they operate to screen messages.

Related documentation:

-

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.
- 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. 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 (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) 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 both Knowledge Manager and 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 `Merge sources checked above` check box.

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

Functions

Screening rules can use three basic functions:

- `Find("")`, where is a text string. It returns the result `true` if the interaction contains the exact string between quotes, ignoring case.
- `RegExFind("")`, where 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("")`, where 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 () or (). 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("", )`
- `RegExFind("",,)`
- `RegExMatch("",`

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")

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	`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 symbol that follows	* 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 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 Knowledge Manager uses Java classes for regular expressions, it is best to consult documents describing the particular version of regular expressions used in Java.