



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 Administrator's Guide

Screening rules

---

## Contents

- 1 More information
- 2 Examples of screening rules
  - 2.1 Example: Credit card number
  - 2.2 Example: North American phone number
  - 2.3 Example: Telltale Words



- Administrator

Screening rules enable you to analyze messages so that you can decide how to handle them. This page describes screening rules and provides examples for common purposes.

### Related documentation:

- 

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.

You can cut, copy, paste, and delete screening rules as well as other eServices Manager objects.

### 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.

### More information

See the following links for more information:

- How to create and test screening rules
- Screening rules reference—This section includes:
  - What screening rules check
  - The functions, arguments, and operators you can use
  - Regular expressions

### Examples of screening rules

---

The following examples show screening rules you can use for typical purposes:

- Credit card number
- North American phone number
- Telltale words

### Example: 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\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\d\d\d\d
```

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

### Example: 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\d))[\s\.\-]?s*\d\d\d[\-\.\s]\d\d\d\d
```

#### Phone Number Regular Expression

Symbols	Meaning	Remarks
\d\d\d	Three digits	

---

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

### Example: 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."