

Complex Match Use in Checks



CHECKCENTRAL



BINARYFORTRESS

Complex Match Use in Checks

Check configuration options provide a variety of approaches when matching incoming email messages and setting their status. This guide uses real-world examples to demonstrate check configurations using complex matches.

For regex use in check configurations, see [Regex Use in Checks](#). For more basic check creation tips, see [Check Creation Guide and Best Practices](#).

Complex Match Examples

Text string with varied numbers, wildcard

In this example we want to match a text phrase which ends in a variable number, in effect a wildcard match for any number beginning with "0." The phrase after the number requires an additional word, ("Incremental"), to complete the match.

To match the full text phrase with the number (and not just the individual words), we put quotations around it:

```
"Server.local Backup-0"
```

Any number after that will adhere to this match since, in this example, they all begin with "0" (e.g. 04120). To include the appending word, we use the plus symbol (+). Quotations are not necessary for a single word: `+Incremental` The space between the phrase and the word allows for other content between the matching content.

Here's how it would look in the check configuration area:

The screenshot shows a configuration interface with two checked options: "Condense Whitespace" and "Combine Attachments". Below these, it states "Message must match" followed by a dropdown menu set to "All" and the text "of the following conditions:". Under this, there are two dropdown menus: "Body Text" and "Complex Match". To the right of "Complex Match" is a text input field containing the regex pattern `"Server.local Backup-0" +Incremental`. To the right of the input field are two small icons: a red trash can and a blue question mark.

Setting status based on status phrase counts

Source: Spanning email

The screenshot shows an email titled "Spanning Backup for Office 365 - Daily Backup Summary for ABC Accounting". The email content displays a summary of backup statuses for various Office 365 services. Each service has a status box with a green dot for successful, a yellow dot for partial, and a red dot for failure, followed by the count and percentage. The services listed are:

- 1 successful (100%)
- 0 partial (0%)
- 0 failure (0%)
- 1 successful (100%)
- 0 partial (0%)
- 0 failure (0%)
- 0 successful (0%)
- 0 partial (0%)
- 0 failure (0%)
- 1 successful (100%)
- 0 partial (0%)
- 0 failure (0%)
- 0 successful (0%)
- 0 partial (0%)
- 0 failure (0%)

In this example, Spanning backup software sends similar messages for each type of backup it makes (email, calendar, drive, etc.), listing the number of success, failure, and partial backups under each backup type.

We can use the consistent backup numbers along with their statuses to match on multiples of certain status phrases, (i.e. "0 failure (0%)"). When all backup types are successful, the email will have five instances of the phrase "0 failure (0%)."

To check the presence of all these phrases we create a rule with the phrase and the number of times we'll see the phrase in the body text. That complex match will look like this: `"0 failure (0%)" {5}`

In the success check area, it will look like this:

Success Criteria

Rules [+ Add Success Rule](#)

- ☒ Condense Whitespace
- ☒ Combine Attachments

Message must match All of the following conditions:

Body Text Complex Match `"0 failure (0%)" {5}`

For the warning status, we're going to do somewhat the opposite. We want to match when we don't see the correct total phrases of "0 partial." If all is successful, the total number of "0 partial" phrasings in the email would be five. The complex match format of that would be: `"0 partial"{5}`

To match the opposite of that, we add a minus symbol (-). The complex match format is then: `-"0 partial"{5}`

Here's what that looks like in the warning area of our check:

Warning Criteria

Rules [+ Add Warning Rule](#)

☒ Condense Whitespace

☒ Combine Attachments

Message must match All of the following conditions:

Body Text

Complex Match

`-"0 partial"{5}`



Complex matches can be tested within CheckCentral by configuring your check and clicking the "Test" button (in the Save section when in wizard mode). This requires existing emails to test against, however. For better understanding of complex match elements, see the guide below.

Complex Match Guide

Simple Matching

A string without quotes, brackets, or mustaches ({ and }) will be treated as a "Match Any Word" when multiple words are specified. To match an exact string, use the equality operator (see below). To match text that contains invalid characters (such as brackets or special characters), wrap the string in quotes. For example: `"Failures(")`

Advanced Matching

Use the +, OR and - operators to create more advanced search queries. Operators are not case-sensitive, but capitalizing them can improve the readability of your search query.

Operator: +

The + operator will only match if the text contains both of the query terms. For example: `cats +dogs` Example with four terms: `cats +dogs +horses +cows`

Operator: OR

The OR operator is represented as a space between string matches. It will match anything that has at least one of your query terms. For example: `cats dogs` Example with four terms: `cats dogs horses cows`

Operator: -

Using the - operator will match anything that **doesn't** have the query term in it. For example: `-dogs` Example with multiple terms: `-(cats dogs)` or `-cats -dogs`

Using Multiple Operators

Use multiple operators together to form more complex queries. For example: `-cats +dogs fish`

Operator Precedence

When more than one operator is used in a query, - is processed first, then +, then OR. To change the precedence of operators, use brackets to separate different statements in your query, much like how you would in an arithmetic expression.

Brackets may also improve the readability of your queries, and do not affect performance. In this example, dogs and birds will be evaluated first because + has a higher precedence than OR: `cats dogs +birds`

Phrases

Match on exact phrases, even ones that contain operators, by using quotes. This example will match the exact phrase "dogs +cars are great." The + operator is ignored because it is in quotes: `"dogs +cars are great"`

Escaping Quotes

To include quotes within matching text, escape the character with a backslash. The example matches on "dogs" (with the quotes) in the search text: `\ "dogs\ "`

Match Count Operator

To specify the number of times a phrase should be matched, use the match count operator. The amount can be any positive whole number, including zero. This example will match the phrase "dogs" if it is found exactly 3 times: `dogs{3}`

Equality Modifier

To match text exactly, you can use the equality modifier (==). This example will match the phrase "dogs," but will not match things like "dogs and cats": `==dogs`

Word Boundary Modifier

To match specific words in text, use the word boundary modifier. This example will match the phrases "dogs" or "dogs and cats," but will not match things like "hotdogs": `=dogs`

Use other operators with the word boundary and equality modifiers, and wrap terms in quotes to include spaces and other special characters. This example will match any text that is not exactly equal to "cats and dogs": `-=="cats and dogs"`

About CheckCentral

CheckCentral Monitoring consolidates and simplifies backup, system, and software email updates into a clean, graphical dashboard, bringing peace of mind to IT administrators of SMBs, Enterprises, and MSPs.

To learn more about CheckCentral, visit: <https://www.checkcentral.cc>

About Binary Fortress Software

Binary Fortress has spent 17 years in pursuit of one goal: create software to make life easier. Our software ranges from display management and system enhancement utilities to monitoring tools and digital signage. IT administrators, professional gamers, coffee-shop owners, and MSPs all rely on Binary Fortress to make their days better, and their lives easier.

Copyright © 2007-2024 Binary Fortress Software, all rights reserved.

The Binary Fortress logo is a trademark of Binary Fortress Software.

The CheckCentral logo is a trademark of Binary Fortress Software.

Binary Fortress Software
1000 Innovation Drive, Suite 500
Kanata, Ontario, Canada
K2K3E7
<https://www.binaryfortress.com>