# How-To: Write simple sieve scripts



## 1. Syntax

**The syntax of the sieve language is quite simple:**

```
if <condition> {
  action1;
  action2;
  ...
 }
```

1. The <*condition*> has:
   a) The part of the email you want to test (body, header, envelope) AND
   b) The test you want to do against that part to perform the required actions.

2. The *actions:* what you want to do to the email (discard, reject, redirect, fileinto, stop, keep).

### 1a. The part of the email you want to test (body, header, envelope)

Here's a typical SMTP session:

widget.net welcome to Modusmail 4.0.330
**helo domain.com**
250 OK
**mail from: <user@domain.com>**
250 OK
**rcpt to: <jim@widget.net>**
DATA
**from: user@domain.com**
**to: jim@widget.com**
**subject: this is a test**

**Hello, this is a test Email.**
**Do not pass go,**
**Do not collect 200$**

**Goodbye**

.

QUIT

The bold black helo/mailfrom/mailto statements are in the envelope.
The bold green from/to/subject lines are in the header.
The bold brown message content lines are in the body.

**1b. The test you want to do against that part to perform the required actions (between the brace brackets { }).**

We won't list all the tests that you can do here - most of them are described in the Sieve Programming Guide and the Sieve RFC - but below are examples of the most commonly used ones.

The tests most people use are:

**:contains <header element(s) to test> <string(s) to match>**
**:matches <header element(s) to test> <string(s) to match>**
**:raw [body only] <strings to find>**
**:text [body only] <strings to find>**

Examples of **":contains"-**based scripts:

Discard messages from a certain sender based on the header

```
if header :contains "from" "jim@domain.com" { discard; stop; }
```

Discard messages from a certain sender based on the envelope

```
if envelope :contains "from" "jim@domain.com" { discard; stop; }
```

Discard messages containing the word "sex" in the subject line

```
if header :contains "subject" "sex" { discard; stop; }
```

***NOTE: You can often test multiple header elements and multiple strings by putting them in square brackets [ ].***

Discard messages from jim and joe

```
if header :contains "from" ["jim@domain.com","joe@domain.com"] { discard; stop; }
```

Discard message if they are FROM or TO jim

```
if header :contains ["from","to"] "jim@domain.com" { discard; stop; }
```

The **Matches statement** can be used for **wildcards**

Example: A spammer keeps sending messages with a fake username of jim-nnn where *nnn* is a random digit from 000 to 999. To block them using wildcards, you would do, e.g.,:

```
if envelope :matches "from" "*jim-???@*" { discard; stop; }
```

The **RAW directive** lets you look at the message as a single flat file without any parsing of the mime-parts.

Example: Block any message with reference to a certain type of language (ISO-code usually in the mime-part blocks)

```
if body :raw :contains "ISO-1234" { discard; stop; }
```

The **TEXT directive** lets you look through a "cleaned-up / deobfuscated" version of the message. Spammers often put random HTML comments in the text-html part to make it hard to do keyword matching.

Example: <!--- moo ->s<!-- moo ->ex <!-- moo ->.

By using the text directive, the entire email body is cleaned up. In the example above, the word "sex" would have all the comments removed to deobfuscate it.

Therefore, in most cases, you'd want to use the **":text" directive** when scanning for **words inside the body of a message.**

Example: block any message that contains the words "foo", "bar", "moo" or "wibble" in the body

```
if body :text :contains ["foo","bar","moo","wibble"] { discard; stop; }
```

Note that it can be "dangerous" to add small words to body scans: you should put a trailing or leading space. Example:

Instead of "ass", try " ass "," ass." " ass!" (notice the SPACE before the "a"), otherwise it can catch "assume," "ambassador," etc.

**2. Actions**

**What you want to do to the email (discard, reject, redirect, fileinto, stop, keep)**

**Discard Statement**

When you tell the sieve scanner to discard the message, it can: (a) send it to the Quarantine, (b) tag-and-pass it, or (c) delete the message outright. This behavior is controlled in the Modus Console > Anti-Spam > Preferences.

Note that a discarded message may not always follow the scripted behavior as it can be overridden at the domain and user levels (e.g. the server level may be set to quarantine but the domain setting is tag-and-pass).

**Reject Statement**

Reject is very similar to the discard statement; the only difference being that you can specify a string to return a bounce message to the sender.

Example:

```
If header :contains "to" "jim@yourdomain.com" {
```

```
Reject “Sorry, Jim no longer works here”;
Stop;
}
```

### Redirect Statement

The redirect statement forwards the message to a mailbox you specify.

Example: Let’s say you want to intercept any email that’s sent to Jim’s account and forward a copy to his boss (bossguy@yourdomain.com). You could write the following:

```
If header  :contains “to” “jim@yourdomain.com” {
  Redirect “bossguy@yourdomain.com”;
  Keep;
  }
```

Note the use of the keep instead of the stop statement. “Keep” simply tells the scanning engine to keep processing the message as if nothing happened (i.e., continue testing with the other sieve scripts and the SCA). The only drawback of catching messages like this is that ‘bossguy’ will receive all of Jim’s spam since this process happens before the SCA kicks in.

### Fileinto Statement

The fileinto lets you move the actual physical message (the MSG file) to a folder somewhere – this is great if you want to implement some sort of archiving system.

Example:

```
If header :contains [“from”,”to”] “yourdomain.com” {
  Fileinto “c:\\program files\\archiveapp\\inboundmessages”;
  Keep;
  }
```

In the hypothetical example above, we have an application called “archiveapp” that has an inbound folder where you would feed it raw messages for processing (and we assume archiving). The script above copies all the messages FROM and TO yourdomain.com to that inbound archive folder. Note the use of the double backslash: it’s important to use this syntax when supplying an absolute file path.

Another use is for “monitoring” of mailboxes. As an example, some of our customers have been approached by law enforcement officials to monitor the mail traffic of an alleged pedophile. In this case, a fileinto script can be used to capture all the raw mail traffic going to and from the alleged pedophile.

Example:

```
If envelope :contains [“from”,”to”] “pedophile@yourdomain.com” {
  Fileinto “c:\\spyfolder”;
  Keep;
  }
```

In this example, all messages going to and from the individual in question are copied to the spyfolder folder.

### Stop and Keep statements

These statements are pretty much self-explanatory if you've followed all the above examples. STOP tells the scanning engine to stop scanning from this point on: it's really useful after a discard or reject statement. You can use an "empty" stop statement to do whitelisting.

Example:

> If header :contains "from" "user@hotmail.com" { stop; }

This would prevent any message from user@hotmail.com from getting blocked by the SCA.

The Keep statement has a different effect: it basically tells the application "regardless of the previous action, keep the message as if nothing happened and process it through normally."

## 3. Other Sieve tricks

### 3.1. Refined attachment blocking

*Question:*
How do I delete certain types of attachments for specific users while quarantining them for everybody else?

*Answer:*
Use sieve instead of the built-in attachment scanning.

Sieve enables more refined filtering than you would normally get from the standard attachment scanning in Modus. For example, the discard behavior is controlled at the server level; however there is a way to "delete" messages even though the discard behavior is set to Quarantine. All you need to do is redirect messages to a "devnull" mailbox.

Here's a typical situation: Widget ISP has some customers who are receiving a lot of .PIF files, and they would like them be deleted instead of showing up the quarantine. Other customers do want them to be quarantined.

Create a "devnull" mailbox
Remove *.pif from the attachment scanning categories
Create your custom script

* This is an old saying from the Unix world: anything you don't want goes to the bitbucket or is "redirected" to >/dev/null. Well – you can do the same with ModusGate and ModusMail.

ModusMail:

- Create a mailbox called devnull in the pertinent domain.
- Go to the mailbox's general properties.
- Check the option "Don't leave a copy of forwarded message in this mailbox"
- Do NOT enter a forwarding address.

Any messages going to this mailbox will be instantly deleted.

Modusgate:
Since there are no local mailboxes in ModusGate, the idea is a bit more convoluted. ModusGate routes messages to a primary mail server. Now most mail servers do have message expiry capability. All you have to do to simulate a "devnull" mailbox is simply to create a mailbox that has a message expiry of 1 day.

- Remove *.pif from attachment scanning categories
- Go to the Modus Console -> Attachment scanning button
- Select “Forbidden Attachments”.
- Expand the “NORMAL” category.
- Remove *.PIF from there.

Next, create your custom script

Let’s say that widget.com has two users (jim@widget.com and joe@widget.com) who want the *.pif files deleted but everyone else stays the same. This is what your sieve script should look like:

This rule only applies to Jim and Joe:

```
If envelope :contains “to” [“jim@widget.com”, “joe@widget.com”] {
  If attachment :matches “*.pif*” {
    Redirect “devnull@widget.com”;
    Stop;
    }
}
```

This rule applies to everyone else:

```
if attachment :matches “*.pif*” { discard; stop; }
```

Go to the Modus console
Go to Anti-SPAM
Go to Custom Scripts
Create a new script and paste the above lines in it.
Enable the script
Stop/Start Moduscan to have the script take effect immediately.

### 3.2 Targeted blocking

*Question:*
How do I block messages going only to a certain user(s)?

*Answer:*
Create a script where you first check if the message is going to a certain user and then apply the rule that you want to implement for that user (embedded if statements).

**Example1:**

Jim is getting mail from a mailing list. He doesn’t know how to unsubscribe from it and you want to block it for him, but not for anyone else. Let’s say messages from the list always come FROM listname@somedomain.com in the header of the messages.

```
If envelope :contains “to” “jim@yourdomain.com” {
  If header :contains “from” “listname@somedomain.com” { discard; stop; }
  }
```

**Example2:**

You want to filter any message with the word “politician” in the subject line for Jim only:

```
If envelope :contains “to” “jim@yourdomain.com” {
  If header :contains “subject“ “politician” { discard; stop; }
  }
```

### 3.3 Logical AND statements with Sieve

*Question:*
How do I block messages that MUST contain a combination of words?

*Answer:*
Use the ALLOF statement or embedded IF statements

Example: You want to block messages that contain both the words “foo” and “bar” in the body of the messages. There are two ways to do it:

1. With an ALLOF statement:

```
If allof (
  Body :text :contains “foo”,
  Body :text :contains “bar”
  ) { discard; stop; }
```

2. With two embedded IF statements:

```
If body :text :contains “foo” {
  If body :text :contains “bar” { discard; stop; }
  }
```

In either case, you can build very complex rules. For instance, let’s say you want to block messages that contain the words “foo” and “bar” and either “you” or “me”, but not “them”, and the rule applies only for Jim:

Example: with an ALLOF Statement:

```
If allof (
  Header :contains “to” “jim@yourdomain.com”,
  Body :text :contains “foo”,
  Body :text :contains “bar”,
  Body :text :contains [“you”,”me”],
  Not body :text :contains “them”
  ) { discard; stop; }
  }
```

Example: with embedded IF Statements (used in combination with allof). **NOTE:** this is more efficient:

```
if Header :contains “to” “jim@yourdomain.com” {
  if allof (
    Body :text :contains “foo”,
    Body :text :contains “bar”,
    Body :text :contains [“you”,”me”],
    Not body :text :contains “them”
    ) { discard; stop; }
}
```

### 3.4 Blocking attachments of a certain type & size

*Question:*
How do I block certain types of attachments over/under a certain size?

*Answer:*
By using the "size" statement.

Example:
You want to block .zip files (which are not normally blocked by attachment scanning) that are over 1 meg in size.

```
if attachment :contains ".zip" {
   if size :over 1000K { discard; stop; }
   }
```

### 3.5 Monitoring attachments going OUT from a domain

*Question:*
I host a corporate domain and they would like to know if any employees are sending out large Emails (perhaps containing attachments) to the outside world. Their security person would like us to keep copies of any large messages going anywhere to the outside world without the sender's knowledge [prevent corporate espionage]. Can it be done?

*Answer:*
Yes - aside from the legal issues involved which we will not get into - it can be done with sieve.

Example:
Let's say you want to capture any message >1 megabyte and send a copy over to a monitoring mailbox without affecting the delivery. This is how you would do it …

```
if envelope :contains "from" "mycorpdomain.com" {
    if size :over 1000K {
       redirect "monitoring@mycorpdomain.com";
       keep;
       }
}
```

In plain English, the above means: if the message is from anybody at "mycorpdomain.com" and if it's over 1 meg in size, send a copy to monitoring@mycorpdomain.com and keep processing as if nothing happened.

### 3.6 Monitoring all email traffic for a specific mailbox

*Question:*
I was asked by authorities to capture all mail traffic coming in and out of a certain mailbox without their knowledge [subpoena]. How can I do this?

*Answer:*
Use a redirect/keep to capture everything.

Example:

You need all mail going to and from jim@yourdomain.com to be captured in a monitoring mailbox.

```
If envelope :contains [“from”,”to”] “jim@yourdomain.com” {
  redirect “monitoring@yourdomain.com”;
  keep;
  }
```

### 3.7 Blocking messages that don’t have a subject line

*Question:*
How do I block messages that don’t have a subject line?

*Answer:*
See script below

```
if allof (
not header :contains "subject" "a",
not header :contains "subject" "e",
not header :contains "subject" "i",
not header :contains "subject" "o",
not header :contains "subject" "u",
not header :contains "subject" "y",
not header :contains "subject" ["!","?"],
not header :contains "subject" "0",
not header :contains "subject" "1",
not header :contains "subject" "2",
not header :contains "subject" "3",
not header :contains "subject" "4",
not header :contains "subject" "5",
not header :contains "subject" "6",
not header :contains "subject" "7",
not header :contains "subject" "8",
not header :contains "subject" "9"
) { discard; stop; }
```

Explanation: If the message doesn’t contain any vowels, punctuation or digits, then discard.

### 3.8 Implementing policy management with sieve

*Question:*
I want to block outgoing messages for some users, nor do I want them to receive emails from the outside world. I only want them to be able to communicate with people internally. How do I do that?

*Answer:*
Creatively using reject statements.

Example:
Let’s say you have a corporate domain (corpdomain.com) and the user betty@corpdomain.com should only be talking to other people @corpdomain.com. She shouldn’t get any emails from the outside world or send emails TO the outside world.

1. Block inbound messages from the outside world to Betty:

```
If not envelope :contains '"from" "corpdomain.com" {
  If envelope :contains "to" "betty@corpdomain.com" {
    reject "Betty is not allowed to receive Email from the outside.  Please contact
sales@corpdomain.com";
    stop;
  }
}
```

2. Block any messages going to the outside world from Betty:

```
If envelope :contains "from" "betty@corpdomain.com" {
  If not envelope :contains "to" "corpdomain.com" {
    reject "You are not allowed to send email to the outside from this address";
    stop;
  }
}
```