

# Exim Virtual Domains with wildcards

Ferry Boender <ferry DOT boender AT electricmonk . nl> version 0.1, November 7, 2008

## 1. Introduction

Often email for multiple domain-names is handled by the same mail server. On a Debian system however, the default configuration does not allow the same user names or the same aliases for different domains. Instead, mail for what is known as a local user (an actual Unix account on the system) is always delivered to that user, no matter the domain it was sent to. Aliases can be set up, but the same restrictions apply to those as well.

This document outlines an Exim4 configuration for Debian which allows you to have complete control over where mail for each recipient address goes, without local user accounts getting in the way. For example, given the following **real** users on the system:

- john (/home/john)
- pete (/home/pete)

You can still do the following:

- Deliver mail for **john@peterson.com** to local user **pete**
- Deliver mail for **john@johnson.com** to local user **john**
- Deliver mail for **root@johnson.com** to remote mail address **john.johnson@gmail.com**
- Deliver mail for **root@peterson.com** to remote mail address **pete.peterson@gmail.com**
- Deliver all other mail for **peterson.com** to local user **pete**
- Deliver all other mail for **johnson.com** to local user **john**

Note that this document assumes Debian GNU/Linux with Exim4 configured with split configuration files. The routers may still be of use on other Operating Systems capable of running Exim4 though. If your Debian system does not have split configuration files, you can enable them by running the command:

### Warning

This will overwrite your current configuration.

```
# dpkg-reconfigure exim4-config
```

## 2. Setup

### 2.1. Configure domains

Okay, first things first. Let's add all the domains for which we want to handle mail to the configuration.

Edit `/etc/exim4/update-exim4.conf.conf`, and modify the `dc_other_hostnames` to include all the domains for which you want to handle mail. On our case, we want to handle mail for **johnson.com** and **peterson.com**:

```
dc_other_hostnames=' johnson.com:peterson.com'
```

Also make sure you've got the `dc_use_split_config` setting set to *true*:

```
dc_use_split_config='true'
```

### 2.2. Adding a router

Next, we must tell Exim4 how it should handle mail for these domains. The default setup is not enough, so what we need to do is add a router to the Exim4 configuration. All the email Exim4 receives (and for which it is configured to handle email in the setup above), goes through all the routers until Exim has found where it can actually deliver the mail. This could be a local user, or a remote mail address, or whatever.

So we add a router to the Exim4 configuration which will do what we want. The router must be referenced by Exim4 before the System aliases are tried, and after the `real_local` router is checked. The `real_local` router allows direct delivery to local users without other processing getting in the way so errors can always be properly delivered)

So we create a new router file: `/etc/exim4/conf.d/router/350_exim4-config_vdom_aliases` with the following contents:

```
vdom_aliases:
    driver = redirect
    allow_defer
    allow_fail
    domains = dsearch;/etc/exim4/virtual
    data = ${expand:${lookup{$local_part}lsearch*@(}/etc/exim4/virtual/$domain)}}
    retry_use_local_part
    pipe_transport = address_pipe
    file_transport = address_file
    no_more
```

This sets up a special custom Exim4 router which will look at any incoming mail and extract the domain-name of the recipient address. The router then looks at the text-files in the `/etc/exim4/virtual/` directory to see if any of the files there are named like the receiving domain. If it finds one, it will look inside that file to see if an alias is listed in it which matches the local part (the part before the `@`). Exim4 then delivers the mail according to what is specified for the alias.

## 2.3. Automatically detect local domains

Normally you'd have to manually specify all the domain-names for which Exim4 should accept and handle email. If you'd like Exim4 to automatically detect which domain-names it should handle email for by looking in the directory `/etc/exim4/virtual/` for domain-names, follow the steps in this section. If you still want to manually configure the local domains, you can skip this step.

### Warning

The instructions in this section modify configuration files which are managed by Debian. Debian does a very poor job of making sure it does not overwrite any configurations you may have edited, so it is possible that whenever Exim4 gets upgraded by Debian, the changes you made in this section are overridden. If you wish to be on the safe side, always make sure you also add any local domain names to the `/etc/exim4/update-exim4.conf.conf` file like you would normally do.

To make Exim4 automatically accept mail for domains found in the `/etc/exim4/virtual/` directory, find the file `/etc/exim4/conf.d/main/01_exim4-config_listmacrosdefs` and edit it. Locate the line that says:

```
domainlist local_domains = MAIN_LOCAL_DOMAINS
```

and change it to:

```
domainlist local_domains = @:localhost:dsearch;/etc/exim4/virtual
```

## 2.4. Create Virtual Domain directory

The last thing we need to do is create the Virtual Domain directory and populate it with alias files:

```
# mkdir -p /etc/exim4/virtual
```

Now we're ready to start adding domains with mail addresses and aliases. Here's an example virtual mail domain alias file for `peterson.com`, which will configure Exim4 as we've discussed at the top of this article:

```
/etc/exim4/virtual/peterson.com  
  
john:pete@localhost  
root:pete.peterson@gmail.com  
*:pete@localhost
```

And the file for +johnson.com:

```
/etc/exim4/virtual/peterson.com  
  
john:john@localhost  
root:john@johnson@gmail.com  
*:john@localhost
```

As you can see the users are specified in the form of:

```
RECIPIENT:USER@HOSTNAME
```

A little explanation of the things we can do in these alias files:

#### RECIPIENT

This is the recipient of the mail as specified in the email. Together with the name of the file (the domain-name), they match the user to which the email was sent. Specifying a \* here will match any recipient not already matched by another RECIPIENT you specified.

#### USER

The part after the : is the user to which to deliver the email. This can be a local user as well as a remote user.

#### HOSTNAME

The part after the @ is the hostname to which to deliver the email. You should always supply this, or Exim4 won't know what to do with it. If you specify localhost here, Exim4 will try to deliver the mail to the user local to the mail system. If you want email to be delivered to a local user, you *must* specify the recipient as @localhost.

Other things we can configure in these files are:

```
# Make any delivery for the recipient fail. Exim4 will send a notification back to the sender  
spam: :fail: This address has been blocked due to spam (this doesn't mean your email is con  
  
# Include a list of other recipients. Exim4 will send the message to all the users in the i  
devlist :include:/etc/exim4/lists/devlist  
  
# Discard all messages send to this recipient:  
noreply :blackhole:
```

## 2.5. Apply configuration and restarting

Finally, we need to apply the new configuration and restart Exim4:

```
# update-exim4.conf
# /etc/init.d/exim4 restart
```

## 3. Testing

An important step in our setup is testing whether mail is delivered correctly. We can do this easily by running Exim4 in address testing mode to see where a message for a recipient will be delivered by Exim4. To start Exim4 in testing mode, run the following command:

```
# exim4 -bt
>
```

You will be put at a prompt where you can enter an recipient email address. Exim4 will then show you where it will be delivered.

The default testing mode doesn't give too much useful output, so we can add the `-d` switch to get more debugging output. Exim4 will now also show us which directives and routers it's going through, and where the recipient email address matches a particular router and the resulting action.

```
# exim4 -bt -d
Exim version 4.63 uid=0 gid=0 pid=22650 D=fb95cfd
Berkeley DB: Sleepycat Software: Berkeley DB 4.3.29: (September 6, 2005)
Support for: crypteq iconv() IPv6 GnuTLS move_frozen_messages
Lookups: lsearch wildlsearch nwildlsearch iplsearch cdb dbm dbmz dsearch nis nis0 passwd
Authenticators: cram_md5 plaintext
Routers: accept dnslookup ipliteral manualroute queryprogram redirect
Transports: appendfile/maildir/mailstore autoreply lmtp pipe smtp
Fixed never_users: 0
Size of off_t: 8
changed uid/gid: forcing real = effective
  uid=0 gid=0 pid=22650
  auxiliary group list: <none>
seeking password data for user "uucp": cache not available
getpwnam() succeeded uid=10 gid=10
configuration file is /var/lib/exim4/config.autogenerated
log selectors = 00000ffc 00189001
trusted user
admin user
seeking password data for user "mail": cache not available
getpwnam() succeeded uid=8 gid=8
user name "root" extracted from gecost field "root"
originator: uid=0 gid=0 login=root name=root
```



obstruction as to the further distribution of this document. You may not ask a fee for the contents of this document, though a fee to compensate for the distribution of this document is permitted.

Modifications to this document are permitted, provided that the modified document is distributed under the same license as the original document and no copyright notices are removed from this document. All contents written by an author stays copyrighted by that author.

Failure to comply to one or all of the terms of this license automatically revokes your rights granted by this license

All brand and product names mentioned in this document are trademarks or registered trademarks of their respective holders.