NAME
postfix-wrapper – Postfix multi-instance API

DESCRIPTION
Postfix versions 2.6 and later provide support for multiple Postfix instances. Instances share executable files and documentation, but have their own directories for configuration, queue and data files.

This document describes how the familiar "postfix start" etc. user interface can be used to manage one or multiple Postfix instances, and gives details of an API to coordinate activities between the postfix(1) command and a multi-instance manager program.

With multi-instance support, the default Postfix instance is always required. The config_directory parameter's default value specifies that instance's configuration file location.

GENERAL OPERATION
Multi-instance support is backwards compatible: when you run only one Postfix instance, commands such as "postfix start" will not change behavior at all.

Even with multiple Postfix instances, you can keep using the same postfix commands in boot scripts, upgrade procedures, and other places. The commands do more work, but humans are not forced to learn new tricks.

For example, to start all Postfix instances, use:

        # postfix start

Other postfix(1) commands also work as expected. For example, to find out what Postfix instances exist in a multi-instance configuration, use:

        # postfix status

This enumerates the status of all Postfix instances within a multi-instance configuration.

MANAGING AN INDIVIDUAL POSTFIX INSTANCE
To manage a specific Postfix instance, specify its configuration directory on the postfix(1) command line:

        # postfix -c */path/to/config_directory command*

Alternatively, the postfix(1) command accepts the instance's configuration directory via the MAIL_CONFIG environment variable (the -c command-line option has higher precedence).

When no Postfix instance information is specified, the postfix(1) command will operate on all Postfix instances.

ENABLING POSTFIX(1) MULTI-INSTANCE MODE
By default, the postfix(1) command operates in single-instance mode. In this mode the command invokes the postfix-script file directly (currently installed in the daemon directory). This file contains the commands that start or stop one Postfix instance, that upgrade the configuration of one Postfix instance, and so on.

When the postfix(1) command operates in multi-instance mode as discussed below, the command needs to execute start, stop, etc. commands for each Postfix instance. This multiplication of commands is handled by a multi-instance manager program.

Turning on postfix(1) multi-instance mode goes as follows: in the default Postfix instance's main.cf file, 1) specify the pathname of a multi-instance manager program with the multi_instance_wrapper parameter; 2) populate the multi_instance_directories parameter with the configuration directory pathnames of additional Postfix instances. For example:

        /etc/postfix/main.cf:
            multi_instance_wrapper = $daemon_directory/postfix-wrapper

multi_instance_directories = /etc/postfix-test

The $daemon_directory/postfix-wrapper file implements a simple manager and contains instructions for creating Postfix instances by hand. The postmulti(1) command provides a more extensive implementation including support for life-cycle management.

The multi_instance_directories and other main.cf parameters are listed below in the CONFIGURATION PARAMETERS section.

In multi-instance mode, the postfix(1) command invokes the $multi_instance_wrapper command instead of the postfix-script file. This multi-instance manager in turn executes the postfix(1) command in single-instance mode for each Postfix instance.

To illustrate the main ideas behind multi-instance operation, below is an example of a simple but useful multi-instance manager implementation:

```
#!/bin/sh

: ${command_directory?"do not invoke this command directly"}

POSTCONF=$command_directory/postconf
POSTFIX=$command_directory/postfix
instance_dirs=`$POSTCONF -h multi_instance_directories |
        sed 's/,/ /'` || exit 1

err=0
for dir in $config_directory $instance_dirs
do
  case "$1" in
  stop|abort|flush|reload|drain)
    test "`$POSTCONF -c $dir -h multi_instance_enable`" \
      = yes || continue;;
  start)
    test "`$POSTCONF -c $dir -h multi_instance_enable`" \
      = yes || {
      $POSTFIX -c $dir check || err=$?
      continue
    };;
  esac
  $POSTFIX -c $dir "$@" || err=$?
done

exit $err
```

PER-INSTANCE MULTI-INSTANCE MANAGER CONTROLS

Each Postfix instance has its own main.cf file with parameters that control how the multi-instance manager operates on that instance. This section discusses the most important settings.

The setting "multi_instance_enable = yes" allows the multi-instance manager to start (stop, etc.) the corresponding Postfix instance. For safety reasons, this setting is not the default.

The default setting "multi_instance_enable = no" is useful for manual testing with "postfix -c *path/name* start" etc. The multi-instance manager will not start such an instance, and it will skip commands such as "stop" or "flush" that require a running Postfix instance. The multi-instance manager will execute commands such as "check", "set-permissions" or "upgrade-configuration", and it will replace "start" by "check" so that problems will be reported even when the instance is disabled.

MAINTAINING SHARED AND NON-SHARED FILES

Some files are shared between Postfix instances, such as executables and manpages, and some files are per-instance, such as configuration files, mail queue files, and data files. See the NON-SHARED FILES section below for a list of per-instance files.

Before Postfix multi-instance support was implemented, the executables, manpages, etc., have always been maintained as part of the default Postfix instance.

With multi-instance support, we simply continue to do this. Specifically, a Postfix instance will not check or update shared files when that instance's config_directory value is listed with the default main.cf file's multi_instance_directories parameter.

The consequence of this approach is that the default Postfix instance should be checked and updated before any other instances.

MULTI-INSTANCE API SUMMARY

Only the multi-instance manager implements support for the multi_instance_enable configuration parameter. The multi-instance manager will start only Postfix instances whose main.cf file has "multi_instance_enable = yes". A setting of "no" allows a Postfix instance to be tested by hand.

The postfix(1) command operates on only one Postfix instance when the -c option is specified, or when MAIL_CONFIG is present in the process environment. This is necessary to terminate recursion.

Otherwise, when the multi_instance_directories parameter value is non-empty, the postfix(1) command executes the command specified with the multi_instance_wrapper parameter, instead of executing the commands in postfix-script.

The multi-instance manager skips commands such as "stop" or "reload" that require a running Postfix instance, when an instance does not have "multi_instance_enable = yes". This avoids false error messages.

The multi-instance manager replaces a "start" command by "check" when a Postfix instance's main.cf file does not have "multi_instance_enable = yes". This substitution ensures that problems will be reported even when the instance is disabled.

No Postfix command or script will update or check shared files when its config_directory value is listed in the default main.cf's multi_instance_directories parameter value. Therefore, the default instance should be checked and updated before any Postfix instances that depend on it.

Set-gid commands such as postdrop(1) and postqueue(1) effectively append the multi_instance_directories parameter value to the legacy alternate_config_directories parameter value. The commands use this information to determine whether a -c option or MAIL_CONFIG environment setting specifies a legitimate value.

The legacy alternate_config_directories parameter remains necessary for non-default Postfix instances that are running different versions of Postfix, or that are not managed together with the default Postfix instance.

ENVIRONMENT VARIABLES

MAIL_CONFIG

When present, this forces the postfix(1) command to operate only on the specified Postfix instance. This environment variable is exported by the postfix(1) -c option, so that postfix(1) commands in descendant processes will work correctly.

CONFIGURATION PARAMETERS

The text below provides only a parameter summary. See postconf(5) for more details.

multi_instance_directories (empty)
An optional list of non-default Postfix configuration directories; these directories belong to additional Postfix instances that share the Postfix executable files and documentation with the default Postfix instance, and that are started, stopped, etc., together with the default Postfix instance.

multi_instance_wrapper (empty)
The pathname of a multi-instance manager command that the **postfix**(1) command invokes when the multi_instance_directories parameter value is non-empty.

multi_instance_name (empty)
The optional instance name of this Postfix instance.

multi_instance_group (empty)
The optional instance group name of this Postfix instance.

multi_instance_enable (no)
Allow this Postfix instance to be started, stopped, etc., by a multi-instance manager.

NON-SHARED FILES

config_directory (see 'postconf -d' output)
The default location of the Postfix main.cf and master.cf configuration files.

data_directory (see 'postconf -d' output)
The directory with Postfix-writable data files (for example: caches, pseudo-random numbers).

queue_directory (see 'postconf -d' output)
The location of the Postfix top-level queue directory.

SEE ALSO

postfix(1) Postfix control program
postmulti(1) full-blown multi-instance manager
$daemon_directory/postfix-wrapper simple multi-instance manager

LICENSE

The Secure Mailer license must be distributed with this software.

AUTHOR(S)

Wietse Venema
IBM T.J. Watson Research
P.O. Box 704
Yorktown Heights, NY 10598, USA