



### Exercise 4.1 Adding a New Startup Service with SysVinit

In this and the following exercise, we will create a simple startup service. First we will do it for a **SysVinit** system. Note that if you are using a **systemd**-based system everything should still work because of the backwards compatibility layer that all distributions utilize. However, in the next exercise we will do natively for **systemd**.

If you are on a **Debian**-based system like **Ubuntu**, make sure you have installed the **sysvinit-utils** and **chkconfig** packages. However, recent versions of **Ubuntu** no longer package **chkconfig**; you'll have to use the **update-rc.d** utility instead.

First we have to create the service-specific script; you can create one of your own for fun, or to get the procedure down just (as root) create a file named `/etc/init.d/fake_service` containing the following content:

```
#!/bin/bash
# fake_service
# Starts up, writes to a dummy file, and exits
#
# chkconfig: 35 69 31
# description: This service doesn't do anything.
# Source function library

. /etc/sysconfig/fake_service

case "$1" in
  start) echo "Running fake_service in start mode..."
        touch /var/lock/subsys/fake_service
        echo "$0 start at $(date)" >> /var/log/fake_service.log
        if [ ${VAR1} = "true" ]
        then
            echo "VAR1 set to true" >> /var/log/fake_service.log
        fi
        echo
        ;;
  stop)
    echo "Running the fake_service script in stop mode..."
    echo "$0 stop at $(date)" >> /var/log/fake_service.log
    if [ ${VAR2} = "true" ]
    then
        echo "VAR2 = true" >> /var/log/fake_service.log
    fi
    rm -f /var/lock/subsys/fake_service
    echo
    ;;
  *)
    echo "Usage: fake_service {start | stop}"
    exit 1
esac
exit 0
```

If you are taking the online self-paced version of this course, the script is available for download from your **Lab** screen.

Make the file above executable and give other proper permissions:

```
$ sudo chmod 755 /etc/init.d/fake_service
```

You'll notice the script includes the file `/etc/sysconfig/fake_service`. (On non-**RHEL** systems you should change this to `/etc/default/fake_service`.) Create it and give it the following contents:

```
VAR1="true"
VAR2="true"
```

Test to see if the script works properly by running the following commands:

```
$ sudo service fake_service  
$ sudo service fake_service start  
$ sudo service fake_service stop
```

Look at the file named `/var/log/fake_service.log`. What does it contain?

For fun you can add additional modes like `restart` to the script file; look at other scripts in the directory to get examples of what to do.

Next we will want to have the ability to start **fake\_service** whenever the system starts, and stop it when it shuts down. If you do:

```
$ sudo chkconfig --list fake_service
```

you will get an error as it hasn't been set up yet for this. You can easily do this with:

```
$ sudo chkconfig --add fake_service
```

and you can turn it on or off at boot time with

```
$ sudo chkconfig fake_service on  
$ sudo chkconfig fake_service off
```

To test this completely you'll have to reboot the system to see if it comes on automatically. You can also try varying the runlevels in which the service is running.