

## Using rsync to update laptop

By Dirk Hart

One of my clients called for help with keeping a database on some of his laptops current. It turns out that he has service personnel in the field repairing appliances. Often enough they need to order a part, but because they need to refer to a manual the ordering was done at the end of the day. Of course the usual thing happens - the service technician forgets what part to order, or confuses one customer with the other and all too often the wrong part gets ordered. Sometimes they try to order an outdated part and the manufacturer can't fill the order. Clearly there is some room for improvement.

Technology to the rescue! Most manufacturers these days provide catalogs on CD-ROM so we installed the application on the laptop and it worked fine. Now the service technician could look up the model, locate the part and order the right thing.

The manufacturers usually supply just one CD-ROM which is inconvenient if you have multiple laptops that need to be updated. We hatched a scheme - could we install the application on a PC and have the data files reside on a RedHat machine running Samba. Samba, as you may know, lets your windows machines share a portion of the disk drive in your RedHat server. Also included with Samba is a nifty utility called rsync ( <http://rsync.samba.org/download.html> ). rsync lets you update a remote machine (eg, laptop) with another machine (eg RedHat server). We now had all the bits of the puzzle in place. Each laptop would connect to the Local Area Network (LAN) and we would use rsync to update the data directory on the laptop.

Rsync has several features that make this an elegant solution. First, you have to authenticate for security purposes. Second, it uses an encrypted data stream both for authentication and for transferring data. Third, rsync can be started by a logon script, by a Windows desktop icon, by the cron scheduler or, in fact, by any other scheme you can think of. Fourth, rsync can also be setup to create new files and delete file when they disappear from the reference machine. Finally - the coolest feature of all - rsync will compare pieces of files on one machine to the corresponding pieces of files on the other machine and only transfer the pieces that are different! Brothers and Sisters that saves a ton of bandwidth and time!!

Because you need an rsync component on each machine (it was already installed on the RedHat machine as it is included with Samba) I located an rsync port for windows at

[http://optics.ph.unimelb.edu.au/help/rsync/rsync\\_pc1.html](http://optics.ph.unimelb.edu.au/help/rsync/rsync_pc1.html) and followed the minimalist approach outlined there. I strayed from the directions and it cost me several hours. Be aware that you will need to login with a username and password that you have predefined on the RedHat machine.

After checking the RedHat machine to make sure ssh was running (I use WebMin) I was ready for some testing. The first time I ran rsetup I had to set up the ssh-keys, but after that it seemed to work fine, so I modified rsetup.bat (it could have been any .bat file ) specific to my situation.

```
echo off
set PATH=c:\rsync;%PATH%
set CYGWIN=binmode tty
set TERM=ansi
set USERNAME=dhart
set HOME=c:\rsync
set RSYNC_RSH=ssh.exe
rsync -aq greatwall:/home/dhart c:\rsync
```

I set the username to match the login I had already set up, set PATH and HOME to match the directory on the laptop where i installed rsync. On the last line I typed in the name of my server (greatwall) and the reference directory and the base target directory.

Also, I modified the parameters a bit. I used -a which ensures that symbolic links, devices, attributes, permissions, ownerships and such are maintained across the transfer. The -z parameter turns on compression, but since I was testing with binary data I didn't bother. Using -v and -vv increases verbosity and -q decreases it. There are many, many more options available at <http://optics.ph.unimelb.edu.au/help/rsync/rsync.html>.

I ran the bat file and entered my password. The first time around things took a long time (it seemed like 10 minutes, but was probably far less) (about 16 image files of 1.3 Megs each), but when i added a couple of file to the reference directory on the RedHat machine just to insert some change into the equation, it took far less time.

Now when the service techs come in after a hard day repairing icemakers and such they just jack into the LAN, double-click the rsync icon and enter their password. They next day they are refreshed and so is the data on their laptops.