NET INTEGRATION TECHNOLOGIES INC.

rsync How-To

How to use rsync to Synchronize Files

September 29, 2004

615-0001-03

How-to use rsync to Synchronize Files

In this How-to you will learn:

- What rsync does
- How to Enable rsync in Nitix™
- How to Configure Rsync:
    - On a Windows® Client
    - As a Windows "Scheduled Tasks" job
    - Between Nitix servers
- Automating rsync with cron in Nitix

What rsync Does

Rsync was designed as a tool for keeping two sets of files on UNIX® platforms to be up-to-date with each other. System administrators have found rsync to be so effective that it has been ported to virtually every OS, including all Microsoft® platforms, Mac® OS X, BSD®, Solaris®, Linux®, and many more.

Rsync works by calculating the differences between a master set of files and a copy of the master files and sending the differences between them in small variable size blocks. Rsync provides uni-directional file and folder synchronization between all TCP/IP compliant operating systems. This would include the ability to synchronize between servers, between a server and client, between a client and a server and even between subfolders on the same system.

The rsync remote-update protocol allows rsync to transfer just the differences between two sets of files across a network connection using an efficient checksum-search algorithm. If the file has only a few small changes, rsync conserves bandwidth by sending only the parts of the file that changed. Conversely, if a given file has changed substantially, the differences synchronized may contain the entire contents of the file.

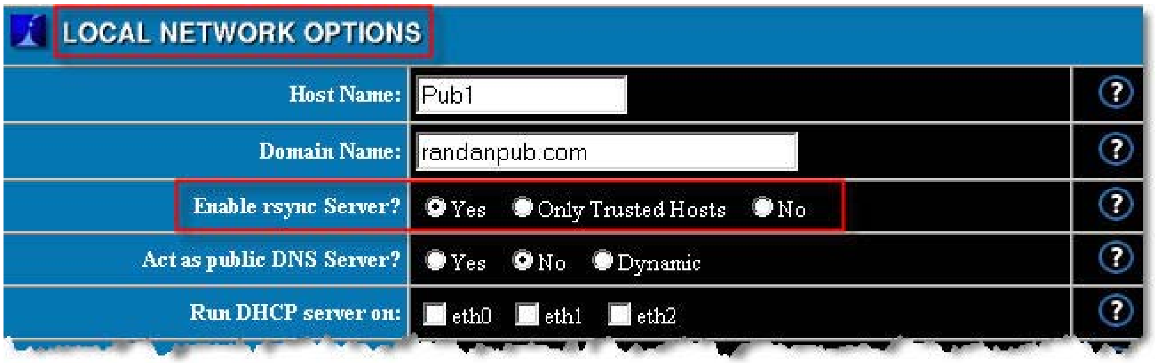
In a Nitix environment rsync can be used with *idb* technology to provide a superior Network backup solution. The three most commonly used rsync implementations to take advantage of *idb* backups of remote systems are backups from:

1. A Windows server to a Nitix server
2. A Windows client to a Nitix server
3. One Nitix server to another Nitix server

Setting up and configuring rsync is the same on either a Windows server or client operating system. We will demonstrate rsync from a Windows XP system to a Nitix server.

Enabling rsync on Nitix

This one is too easy…



LOCAL NETWORK OPTIONS

Host Name: Pub1

Domain Name: randanpub.com

Enable rsync Server? ● Yes ○ Only Trusted Hosts ○ No

Act as public DNS Server? ○ Yes ● No ○ Dynamic

Run DHCP server on: □ eth0 □ eth1 □ eth2

Figure 1 - Enable rsync

Click on "*Yes*" or "*Only Trusted Hosts*" and you're done.

Configuring Rsync

Configuring rsync from a Windows Client

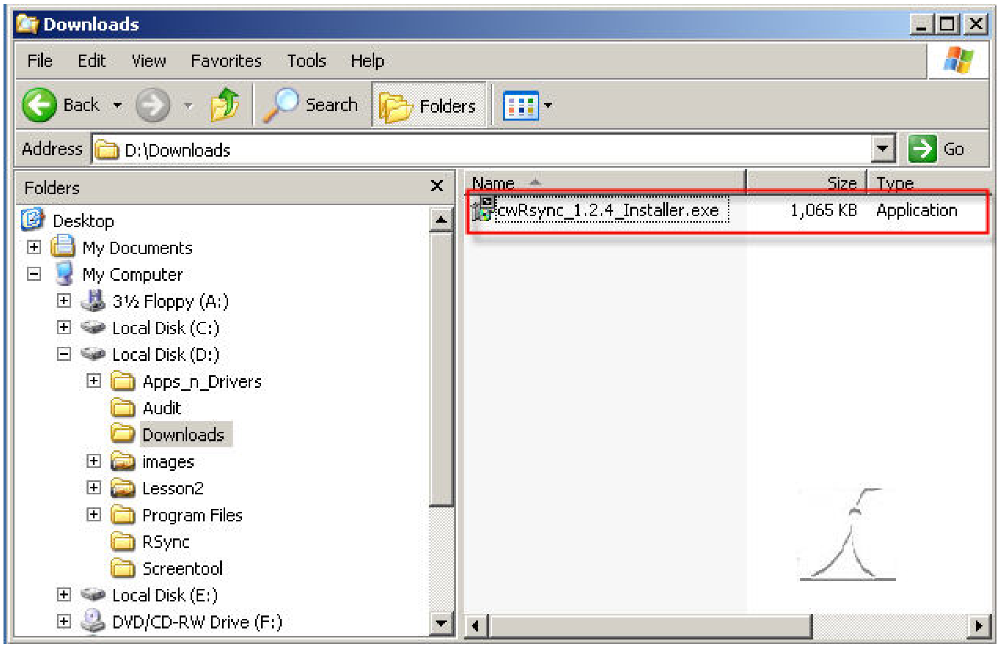
Since rsync is a UNIX-based program, a Windows port or version of the program is required to provide rsync functionality in a Window environment. **cwRsync** is a Windows command line implementation of rsync which provides a scriptable solution that can be automated to run at scheduled times as a Windows Scheduled Tasks job.

Installing cwRsync

cwRsync is s a free 1MB download available from:

http://prdownloads.sourceforge.net/sereds/cwRsync_1.2.5_Installer.zip?download

Choose your closest mirror site and download the zipped file and unzip it to your hard drive. Unzip the file with WinZip® or your favorites unzip program and then double-click on the cwRsyn_latest_version_number_installer.exe file to install the program.

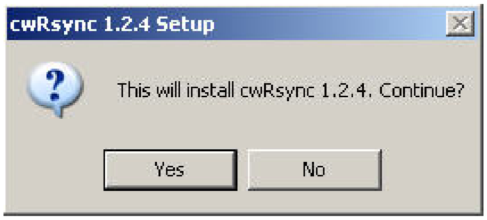


Address D:\Downloads

cwRsync_1.2.4_Installer.exe 1,065 KB Application

Figure 2 - Running the cwRsync executable

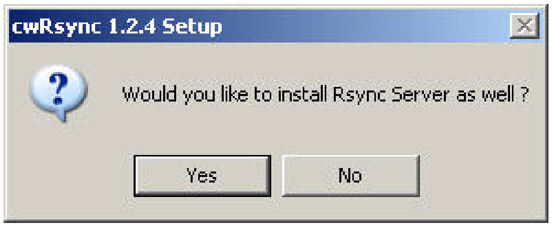
The cwRsync Setup box appears.



This will install cwRsync 1.2.4. Continue?

Figure 3 - cwRsync setup box

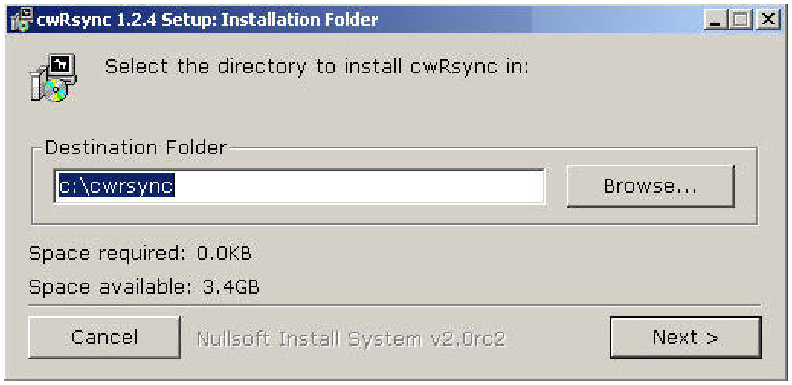
Click yes to continue.



Would you like to install Rsync Server as well ?

Figure 4 - Install rsync server query box.

If you do not require an rsync server on the Windows client, select *No*. In this example the Nitix server hosts the rsync server daemon (Server service) so it is not required on the client.



cwRsync 1.2.4 Setup: Installation Folder

Select the directory to install cwRsync in:

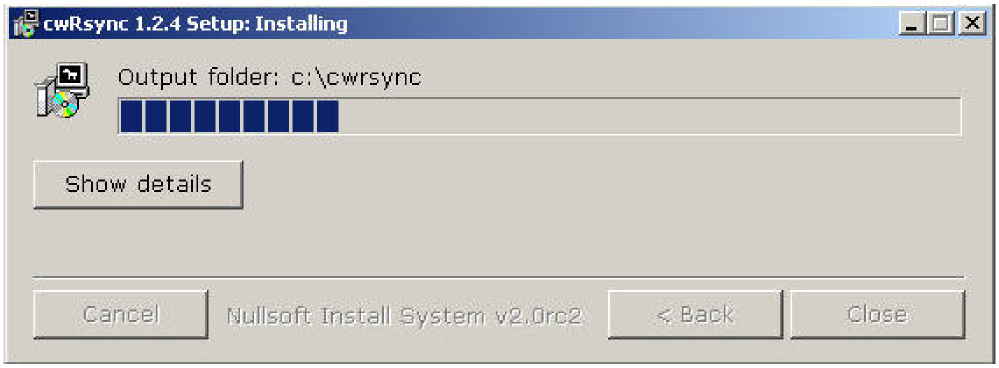
Destination Folder: c:\cwrsync

Space required: 0.0KB

Space available: 3.4GB

Figure 5 – Install folder

Click on *Next* > to accept the default install folder of C:\cwrsync.



cwRsync 1.2.4 Setup: Installing

Output folder: c:\cwrsync

Figure 6 – Install status

This is not a large program so the files will be copied quickly. Click *Close* to continue. That’s it, the installation is complete.

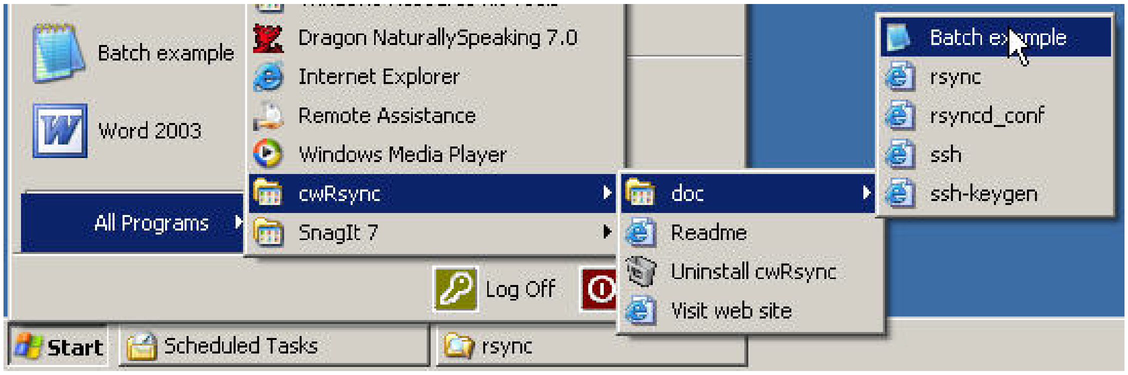
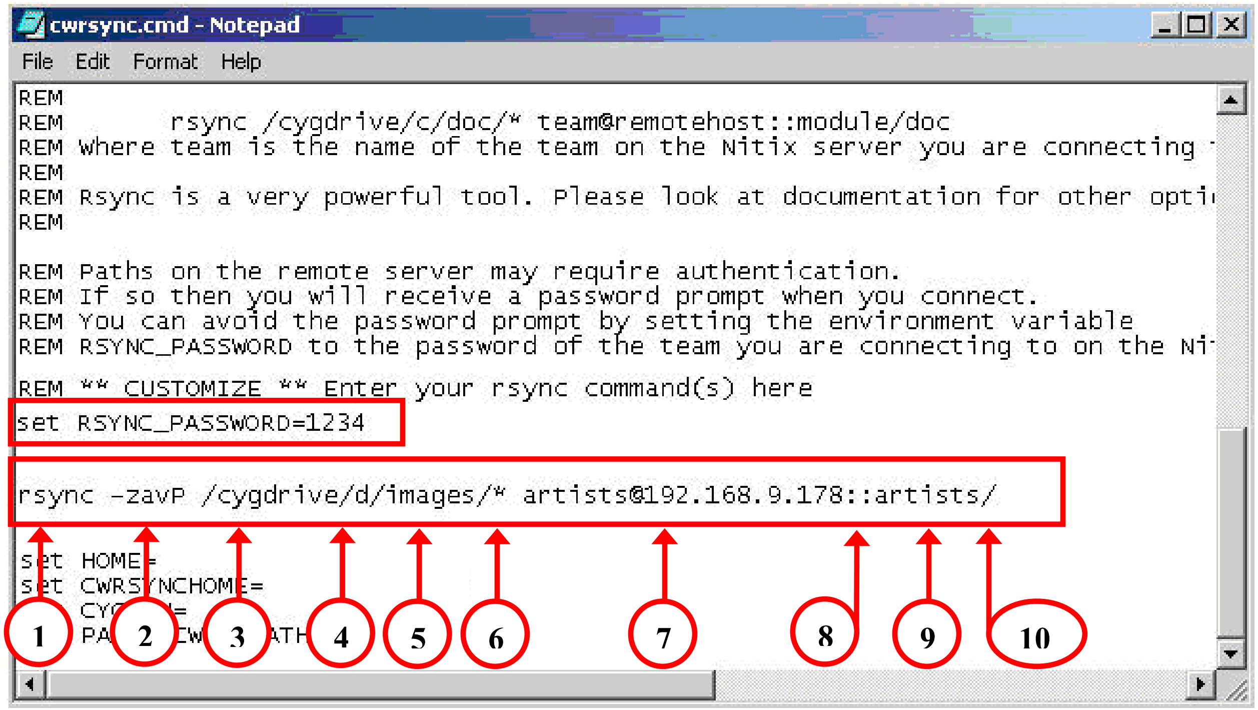


Figure 7 – cwRsync installed

cwRsync will show up on the *All Programs* menu; however there is no program file to run as cwRsync is run from a Batch file.

The Batch example file will need to be edited to configure the environment and run the rsync command for you. This Batch file can also be used with the Windows Schedule Tasks scheduler to run the rsync command at whatever frequency works in your particular environment. Let's look inside the cwRsync Batch file found in C:\cwrsync.

Click on the **Start** > **All Programs** > **cwRsync** > **Doc** > "**Batch example**" menu selection to bring up the Batch file to be edited.



```
REM
REM       rsync /cygdrive/c/doc/* team@remotehost::module/doc
REM Where team is the name of the team on the Nitix server you are connecting 
REM
REM Rsync is a very powerful tool. Please look at documentation for other opti
REM

REM Paths on the remote server may require authentication.
REM If so then you will receive a password prompt when you connect.
REM You can avoid the password prompt by setting the environment variable
REM RSYNC_PASSWORD to the password of the team you are connecting to on the Ni
REM ** CUSTOMIZE ** Enter your rsync command(s) here
set RSYNC_PASSWORD=1234

rsync -zavP /cygdrive/d/images/* artists@192.168.9.178::artists/

set HOME=
set CWRSYNCHOME=
    CY   =
    PA   CW   ATH
```

Figure 8 – The cwrsync.cmd Batch file

The Batch file shown above has been edited by adding the two important lines shown in boxes. The first added line "set RSCYN_PASSWORD=1234" is the password of the team account "artists" on the Nitix server where we will synchronize the images files. Embedding the password here allows the Batch file to execute without halting to wait for a user to enter the password manually. Note that inserting the password here means that the password is stored in plain text and it should be protected accordingly.

The second line is the rsync command itself. This is the command that will do the synchronization from the Windows XP to our Nitix server.

Let’s look at all of the rsync parameters.

rsync Parameters

1. **rsync** This of course is the executable command.
2. **-zavP** Four switches that configure rsync, lets look at –z –a –v –P separately:

   -z, --compress. With this option, rsync compresses any data from the files that it sends to the destination machine. This option is useful on slow connections. The compression method used is the same method that the classic UNIX gzip compression utility uses.

   -a —archive. It is a quick way of saying you want recursion and want to preserve almost everything during the synchronization.

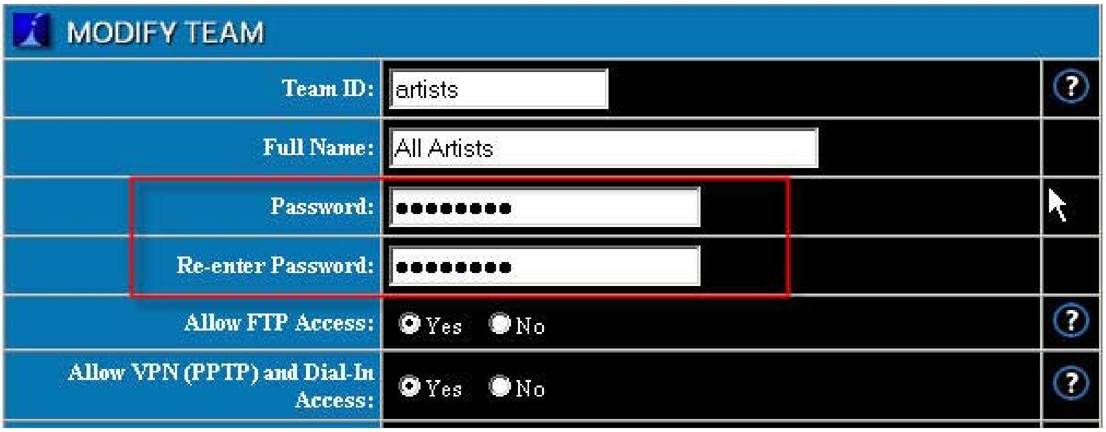
   -v, --verbose. This option increases the amount of information you are given during the transfer. By default, rsync works silently. A single -v will give you information about what files are being transferred and a brief summary at the end. Two -v flags will give you information on what files are being skipped and slightly more information at the end.

   -P, --partial and –progress. The –P (Capital P ;-)) option is equivalent to using both the –partial and the –progress rsync options.

   > **--partial** – Using this option tells rsync to keep a partial file in the event of an interruption of the file transfer. This should make a subsequent transfer of the file faster.
   >
   > **--progress** – Tells rsync to print to the screen information showing the progress of the transfer.
3. **/cygdrive** This parameter is to signal the next parameter which is the local drive letter containing the files and or folders to be synchronized.
4. **/D** This is the local **D:** drive holding the folder to be synchronized.
5. **/images** This is the folder to be synchronized.
6. **/*** This tells rsync to synchronize all files and folders below the images folder to the destination folder.
7. **artists@192.168.9.178** *artists* is the team name that has permission to access the artists folder on the Nitix rsync server at 192.168.9.178. Where “artists” can be any user or team account and the IP address 192.168.9.178 can be replaced by a host name. The RSYNC_PASSWORD variable entered above must match the password of the user or team account entered here.
8. **::** A double colon in the destination field tells rsync to copy from the local server to the remote host. The double colon also separates the hostname from the path
9. **artists** This is the path or folder share located on the rsync server.
10. **/** The forward slash appended to the trailing directory eliminates any confusion rsync might have with the command. Without it, you might end up with /artists/dir/dir/ or something. You may however specify subdirectories below artists as a destination by entering artists/subdir/.

That's the basics of configuring the Batch file to synchronize our files. As noted in the REM section of the Batch file…rsync is a very powerful command, look at the documentation to learn about many more options rsync provides.

Before we copy some files into the D:\images folder let's ensure that the *artists* team on the server is indeed configured with a password (1234) or our Batch job will fail. From the *User Setup* screen in WebConfig, the *artists* team account is shown and we have modified the password to 1234 so the Batch file will work.

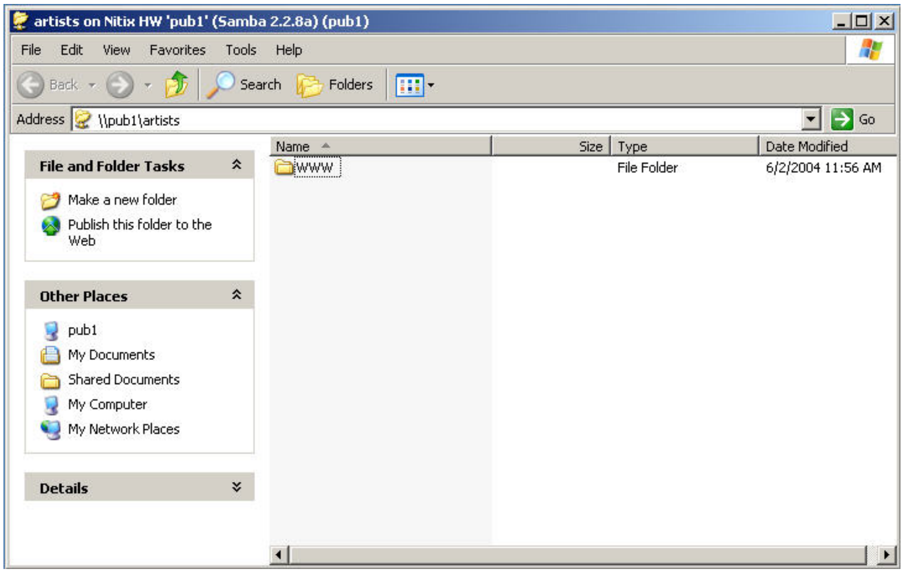


MODIFY TEAM

Team ID:	artists
Full Name:	All Artists
Password:	●●●●●●●●●
Re-enter Password:	●●●●●●●●●
Allow FTP Access:	◉ Yes ○ No
Allow VPN (PPTP) and Dial-In Access:	◉ Yes ○ No

Figure 9 – Configuring a team password

The server configuration is now complete and we are ready to demonstrate rsync. Let's look at the folders on the local folder and on the rsync server before the Batch file is run. We'll start with the *artists* folder on PUB1 which is the 192.168.9.178 address we configured in the Batch file.



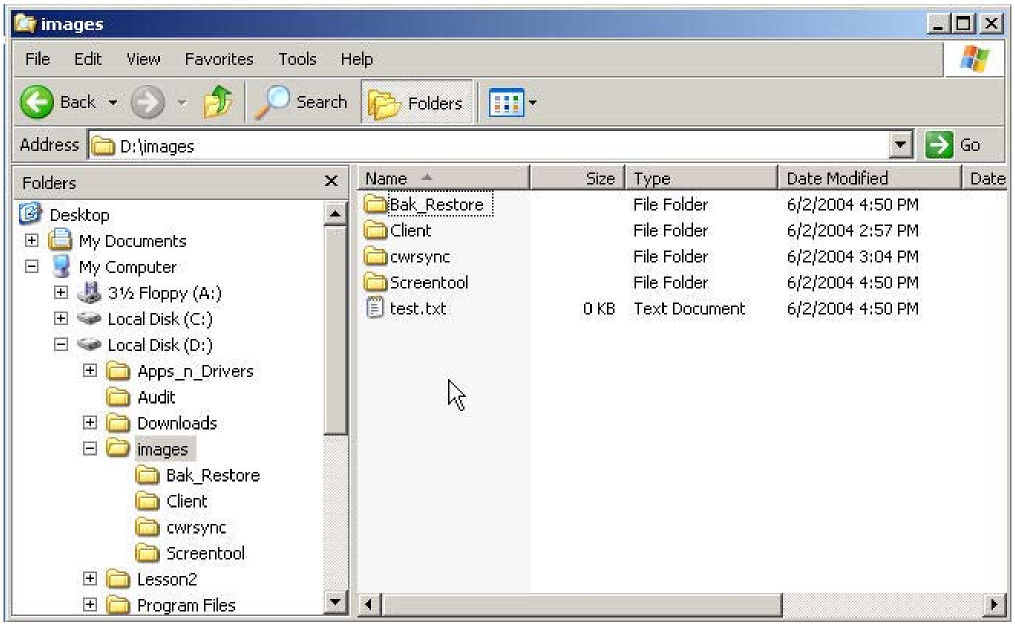
artists on Nitix HW 'pub1' (Samba 2.2.8a) (pub1)

Address \\pub1\artists

Name	Size	Type	Date Modified
www		File Folder	6/2/2004 11:56 AM

Figure 10 – The empty artists folder on Pub1

In the screen capture above we can see that there are no files, except the default WWW folder, on the *artists* share on Pub1. At the client machine:

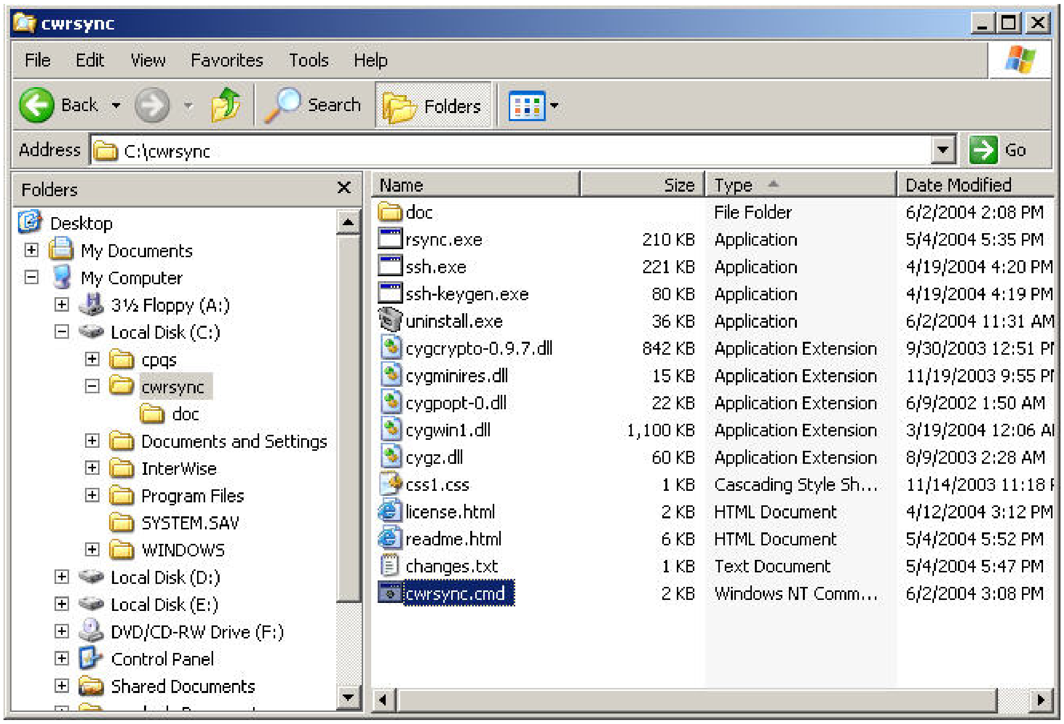


Address: D:\images

Name	Size	Type	Date Modified	Date
Bak_Restore		File Folder	6/2/2004 4:50 PM	
Client		File Folder	6/2/2004 2:57 PM	
cwrsync		File Folder	6/2/2004 3:04 PM	
Screentool		File Folder	6/2/2004 4:50 PM	
test.txt	0 KB	Text Document	6/2/2004 4:50 PM	

Figure 11 – Local D: drive images folder

There are four folders and 1 test.txt file in the D:\images folder.



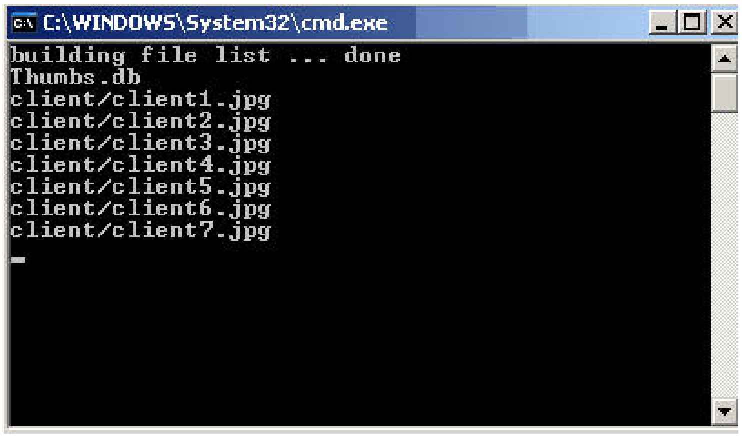
Address: C:\cwrsync

Name	Size	Type	Date Modified
doc		File Folder	6/2/2004 2:08 PM
rsync.exe	210 KB	Application	5/4/2004 5:35 PM
ssh.exe	221 KB	Application	4/19/2004 4:20 PM
ssh-keygen.exe	80 KB	Application	4/19/2004 4:19 PM
uninstall.exe	36 KB	Application	6/2/2004 11:31 AM
cygcrypto-0.9.7.dll	842 KB	Application Extension	9/30/2003 12:51 PI
cygminires.dll	15 KB	Application Extension	11/19/2003 9:55 PI
cygpopt-0.dll	22 KB	Application Extension	6/9/2002 1:50 AM
cygwin1.dll	1,100 KB	Application Extension	3/19/2004 12:06 AI
cygz.dll	60 KB	Application Extension	8/9/2003 2:28 AM
css1.css	1 KB	Cascading Style Sh...	11/14/2003 11:18 I
license.html	2 KB	HTML Document	4/12/2004 3:12 PM
readme.html	6 KB	HTML Document	5/4/2004 5:52 PM
changes.txt	1 KB	Text Document	5/4/2004 5:47 PM
cwrsync.cmd	2 KB	Windows NT Comm...	6/2/2004 3:08 PM

Figure 12 – Launching the Batch file

At the Windows XP client machine navigate to the C:\cwrsync folder and double click on the *cwrsync.cmd* file execute it.

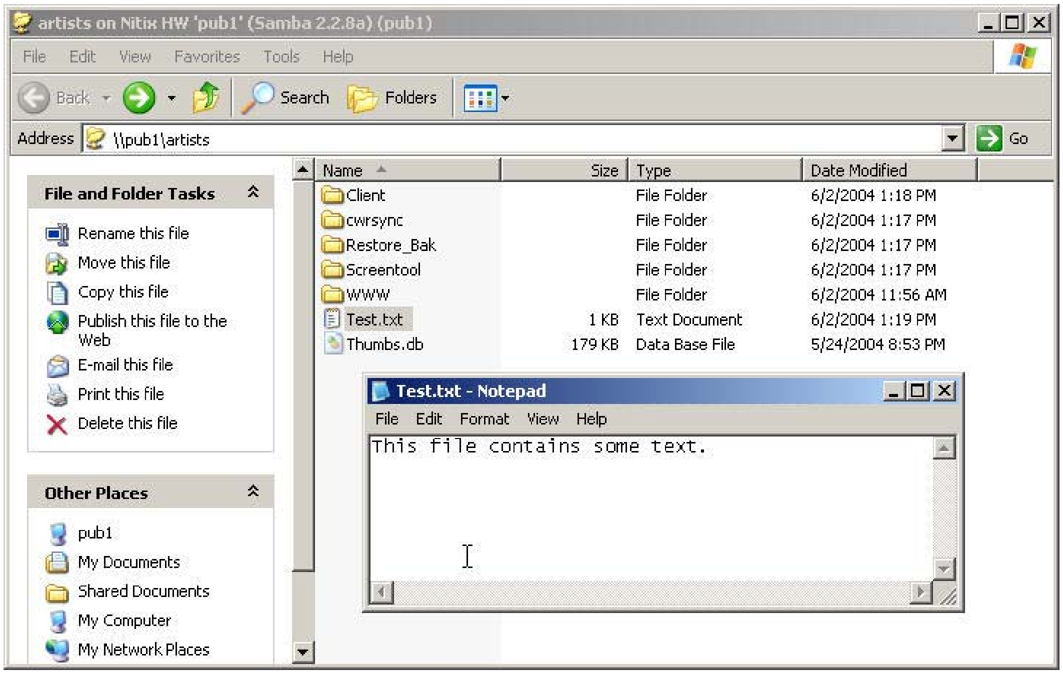
The *cwrsync.cmd* file executes and starts copying files to the *artists* share on 192.168.9.178.



```
building file list ... done
Thumbs.db
client/client1.jpg
client/client2.jpg
client/client3.jpg
client/client4.jpg
client/client5.jpg
client/client6.jpg
client/client7.jpg
```

Figure 13 – The Batch file progress window

When the Batch files completes the command prompt closes. We can now view the *artists* share to confirm that the files were copied over.

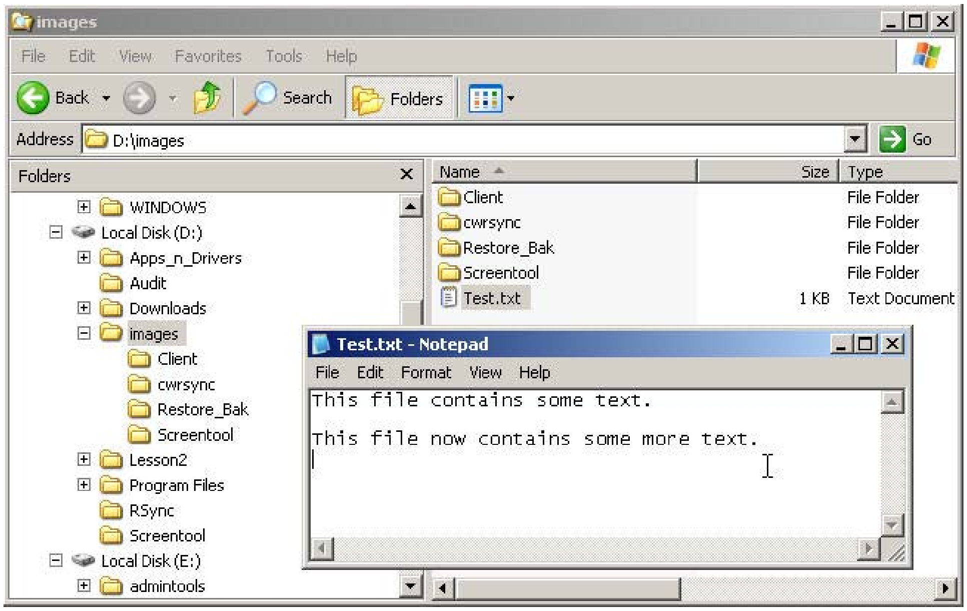


Name	Size	Type	Date Modified
Client		File Folder	6/2/2004 1:18 PM
cwrsync		File Folder	6/2/2004 1:17 PM
Restore_Bak		File Folder	6/2/2004 1:17 PM
Screentool		File Folder	6/2/2004 1:17 PM
WWW		File Folder	6/2/2004 11:56 AM
Test.txt	1 KB	Text Document	6/2/2004 1:19 PM
Thumbs.db	179 KB	Data Base File	5/24/2004 8:53 PM

```
This file contains some text.
```

Figure 14 – The artists folder after rsyncing

In the screen capture shown above the *artists* share at the Pub1 server now contains the files and folders from the image directory on our Windows XP machine. The test.txt file has been opened to demonstrate the file update feature of Rsync. The test.txt file contains the text "this file contains some text”. We will now go back to the client, modify the test.txt file, and then resynchronize the images folder.



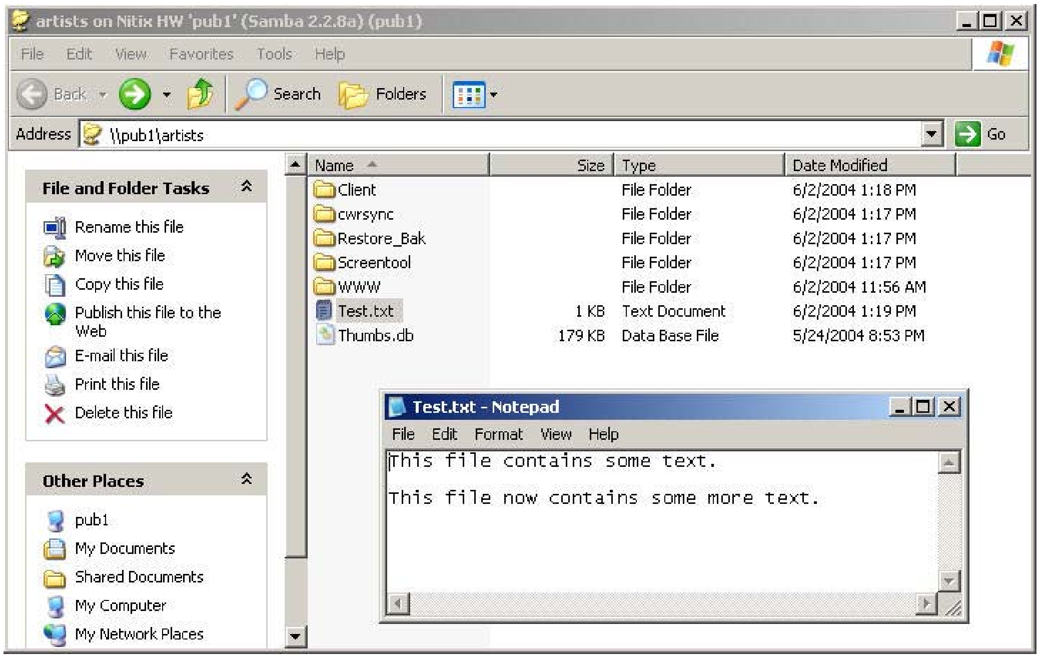
Name	Size	Type
Client		File Folder
cwrsync		File Folder
Restore_Bak		File Folder
Screentool		File Folder
Test.txt	1 KB	Text Document

```
This file contains some text.

This file now contains some more text.
```

Figure 15 – Local File Modification

The screen capture above shows the modified test.txt file in the images directory at the client machine. Test.txt has been opened to show the added text. Save the file, run the *cwrsync.cmd* Batch file again and the file will synchronize. We’ll return to the rsync server to verify that the changes to test.txt succeeded.



Name	Size	Type	Date Modified
Client		File Folder	6/2/2004 1:18 PM
cwrsync		File Folder	6/2/2004 1:17 PM
Restore_Bak		File Folder	6/2/2004 1:17 PM
Screentool		File Folder	6/2/2004 1:17 PM
WWW		File Folder	6/2/2004 11:56 AM
Test.txt	1 KB	Text Document	6/2/2004 1:19 PM
Thumbs.db	179 KB	Data Base File	5/24/2004 8:53 PM

```
This file contains some text.

This file now contains some more text.
```

Figure 16 – Synchronized file modification

At \\pub1\artists, as shown in the address bar, we can see that the test.txt file has been opened and displays the updated text indicating that rsync successfully updated the file.

An Important Note on Paths Containing a Space Character:

If you need to rsync a source folder whose path contains a space then you will need to enclose the path in quotes in a very specific manner. The best way to illustrate this is with an example. If you need to rsync the C:\Documents and Settings\Administrator\My Documents folder to an rsync server the syntax would be:

```
rsync -zavP /cygdrive/c/"Documents and Settings"/Administrator/"My Documents"/*
artists@192.168.9.1::artists/
```

Notice that there are two sets of quotation marks that start and end between the / delimiter in “Documents and Settings” and again in “My Documents”. If you try to enclose the whole path in quotes rsync will fail.

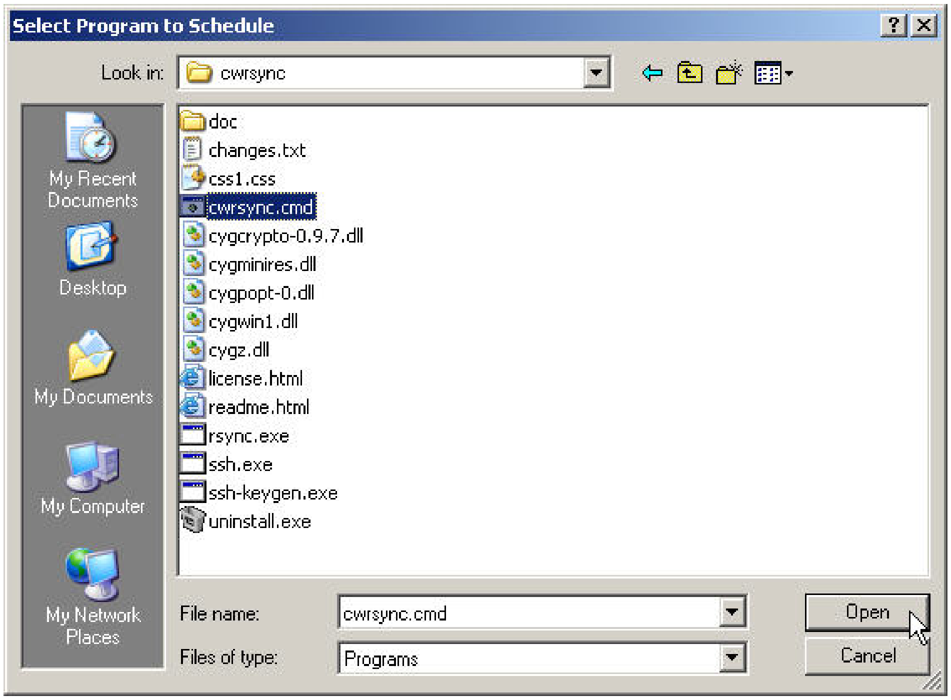
An Important Note on File Permissions:

Using rsync will result in all source NTFS permissions being lost when copying files to a Nitix server. Files will inherit the permissions on the folder where the files have been transferred to. This is a limitation of the rsync utility and not the Samba service.

Automating rsync - Running cwrsync.exe with Windows Schedule Tasks

Now that the rsync server is configured and the client rsync Batch file has been tested all that remains is to automate the regular synchronization of the files between the client and the server. This can be easily accomplished with the Windows Scheduled Tasks Utility.

At the Windows machine open the **control panel** and open the **Scheduled Tasks** utility. Double click the **Add Scheduled Task** icon and click **Next >** to continue. Click on the **Browse…** button in the **Scheduled Task wizard** to locate the cwrsync.cmd file located it the C:\cwrsync folder as shown in the following diagram.



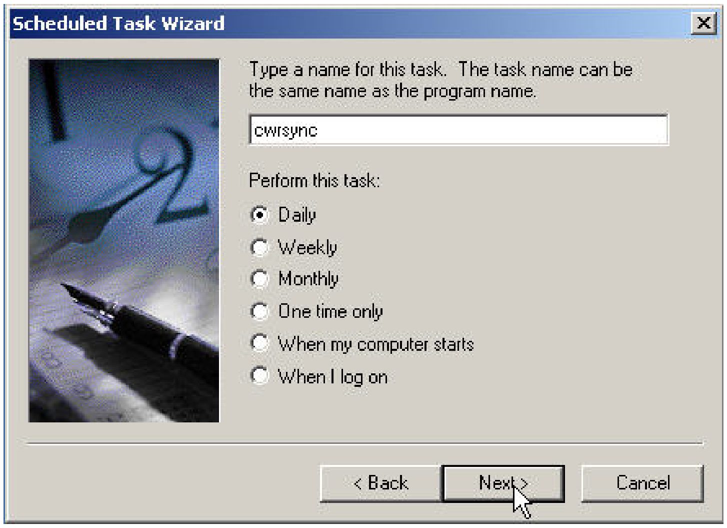
Look in: cwrsync

File name: cwrsync.cmd

Files of type: Programs

Figure 17 – Scheduling the cwrsync.cdm file

Select the cwrsync.cmd file and select the *Open* button to name the task and schedule the frequency the task performs.



Type a name for this task. The task name can be the same name as the program name.

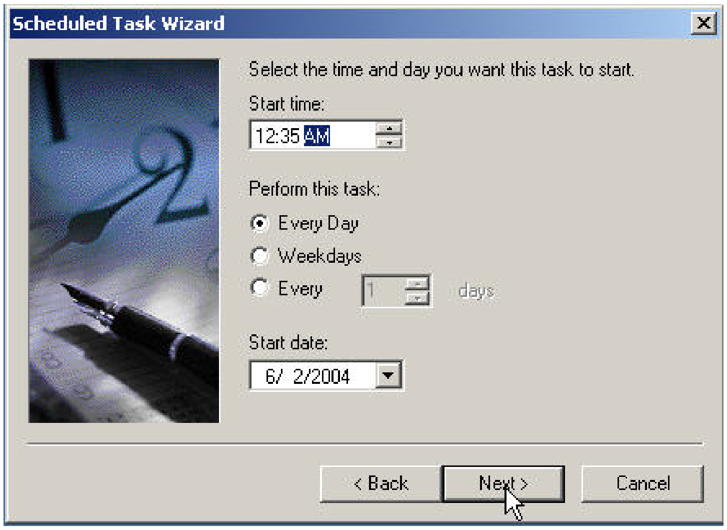
cwrsync

Perform this task:

- ● Daily
- ○ Weekly
- ○ Monthly
- ○ One time only
- ○ When my computer starts
- ○ When I log on

Figure 18 – Set schedule frequency

For this demonstration we have chosen to schedule the job to run Daily.



Scheduled Task Wizard

Select the time and day you want this task to start.

Start time: 12:35 AM

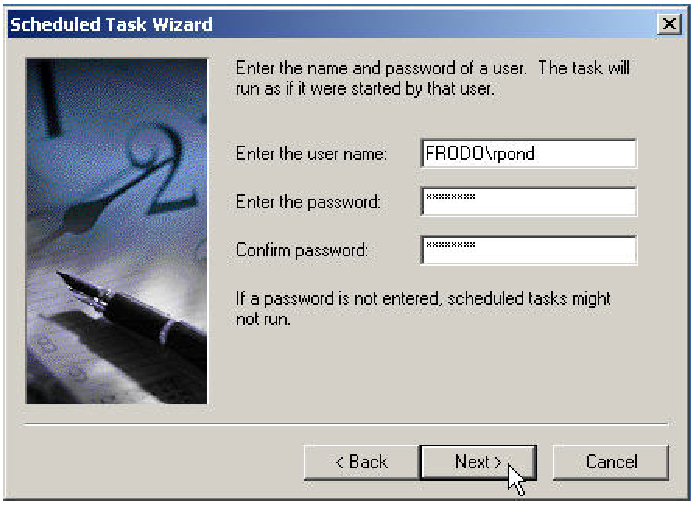
Perform this task:

- ● Every Day
- ○ Weekdays
- ○ Every 1 days

Start date: 6/ 2/2004

Figure 19 – Set the Start time

You can then select the time of day the task runs. Select *Next* > to continue.



Scheduled Task Wizard

Enter the name and password of a user. The task will run as if it were started by that user.

Enter the user name: FRODO\rpond

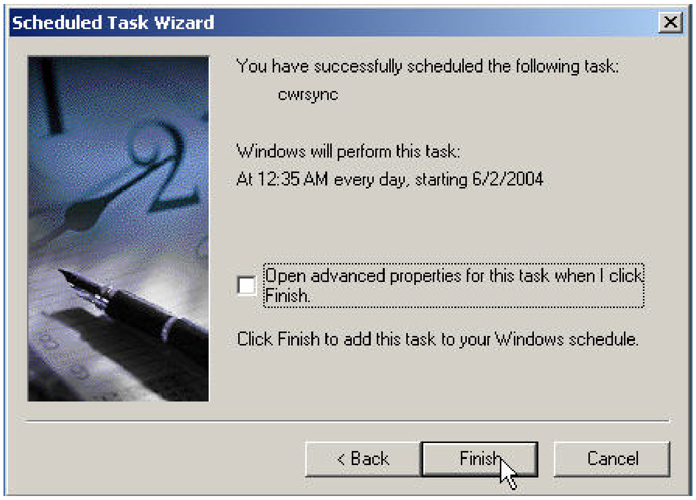
Enter the password: ********

Confirm password: ********

If a password is not entered, scheduled tasks might not run.

Figure 20 – Authentication credentials

Enter the username and password of the user whose authentication credentials you want this task to run as. The cwrsync.cmd or rsync executable do not require administrative privileges to execute.



You have successfully scheduled the following task:
cwrsync

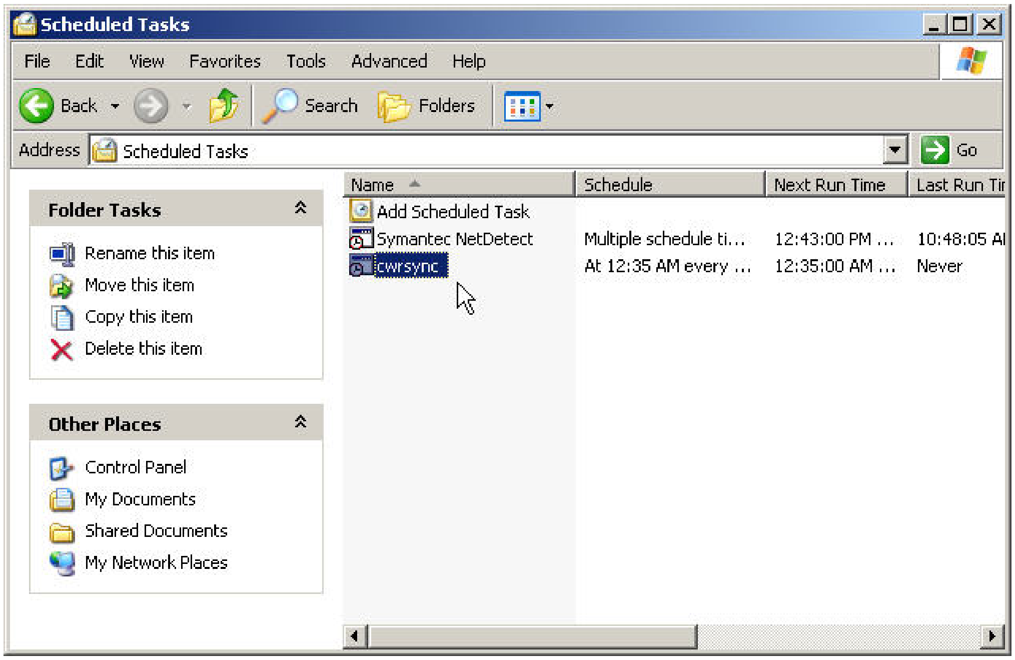
Windows will perform this task:
At 12:35 AM every day, starting 6/2/2004

☐ Open advanced properties for this task when I click Finish.

Click Finish to add this task to your Windows schedule.

Figure 21 – Task Summary box

Click *Finish* to continue.



Name	Schedule	Next Run Time	Last Run Ti
Add Scheduled Task			
Symantec NetDetect	Multiple schedule ti...	12:43:00 PM ...	10:48:05 AI
cwrsync	At 12:35 AM every ...	12:35:00 AM ...	Never

Figure 22 – The Scheduled Task

The new scheduled task shows up in the *Scheduled Tasks* lists and will execute every day at 12:35 AM every day to synchronize the images folder to the “192.168.9.178” Pub1 Nitix server.

Using rsync between Nitix servers

All that’s required to synchronize files between two Nitix servers is to is to have rsync enabled at the server, an understanding of the rsync syntax and a command line prompt.

Rsync was enabled in the Enabling rsync section above, we’ll review the rsync syntax here, and a command prompt will be accessed via putty.exe a free telnet utility.

Putty.exe is widely available on the Internet. You can do a search on putty.exe or try: http://www.tucows.com/preview/195286.html

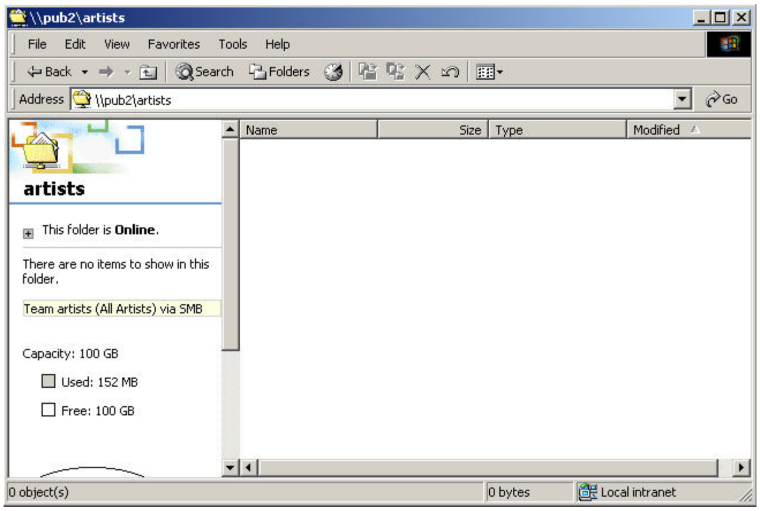
Rsync can be used to push changes out to another file server or pull files from a remote computer back to itself. We will demonstrate:

1. Pushing out files from the /home/artists/Files folder on PUB1 to the /home/artists/Files folder on PUB2.

2. Pulling the files in the /home/artists/Files folder on PUB2 to the /home/dans/Files/ folder on PUB1

Pushing:

The /home//artists/Files folder on PUB2 currently has no files as shown in the screen capture below.



Address: \\pub2\artists

This folder is **Online**.

There are no items to show in this folder.

Team artists (All Artists) via SMB

Capacity: 100 GB

Used: 152 MB

Free: 100 GB

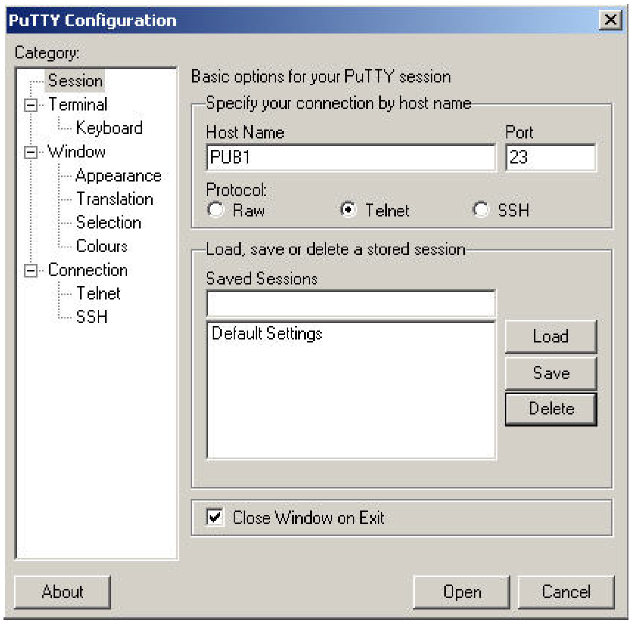
0 object(s)

0 bytes

Local intranet

Figure 23 – Empty artist folder on pub2

We will run rsync from PUB1 and push the file synchronization out to the PUB2 server. To do this, open a telnet session using putty.exe as shown in the screen capture below.



Basic options for your PuTTY session

Specify your connection by host name

Host Name: PUB1

Port: 23

Protocol: ○ Raw ● Telnet ○ SSH

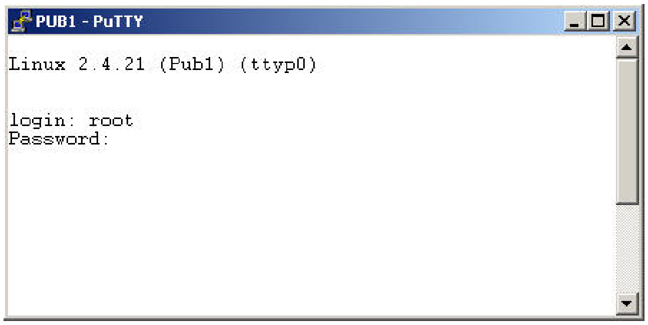
Load, save or delete a stored session

Saved Sessions: Default Settings

☑ Close Window on Exit

Figure 24 – Establishing a Telnet session to pub1

Enter a username and password to authenticate the telnet session.

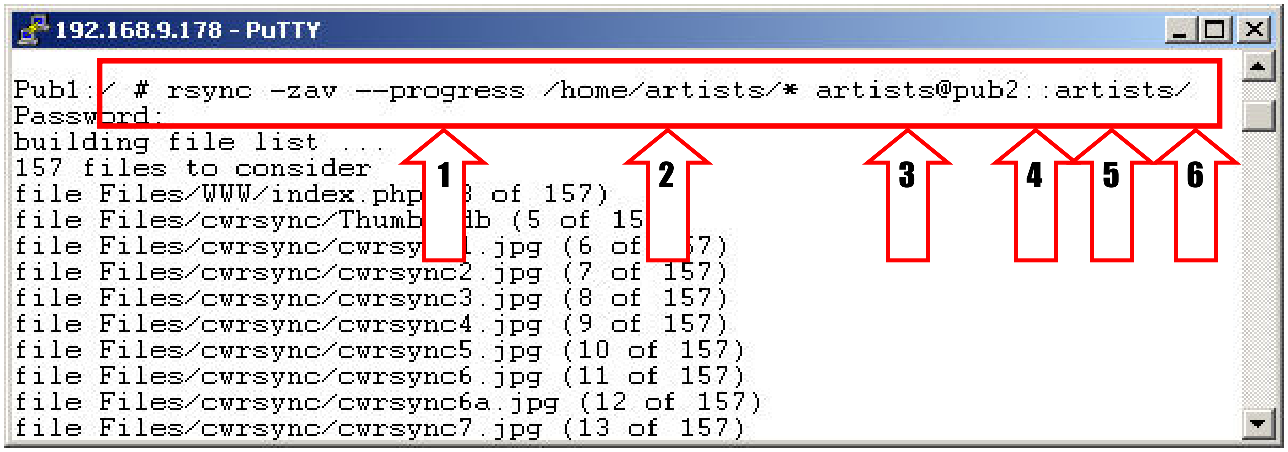


```
Linux 2.4.21 (Pub1) (ttyp0)


login: root
Password:
```

Figure 25 – Telnet Logon credentials

The rsync command to synchronize the files **from** PUB1:/home/artists folder **to** the PUB2/artists folder is shown below.



```
Pub1:/ # rsync -zav --progress /home/artists/* artists@pub2::artists/
Password:
building file list ...
157 files to consider
file Files/WWW/index.php  3 of 157)
file Files/cwrsync/Thumb  db (5 of 15
file Files/cwrsync/cwrsy  1.jpg (6 of  57)
file Files/cwrsync/cwrsync2.jpg (7 of 157)
file Files/cwrsync/cwrsync3.jpg (8 of 157)
file Files/cwrsync/cwrsync4.jpg (9 of 157)
file Files/cwrsync/cwrsync5.jpg (10 of 157)
file Files/cwrsync/cwrsync6.jpg (11 of 157)
file Files/cwrsync/cwrsync6a.jpg (12 of 157)
file Files/cwrsync/cwrsync7.jpg (13 of 157)
```

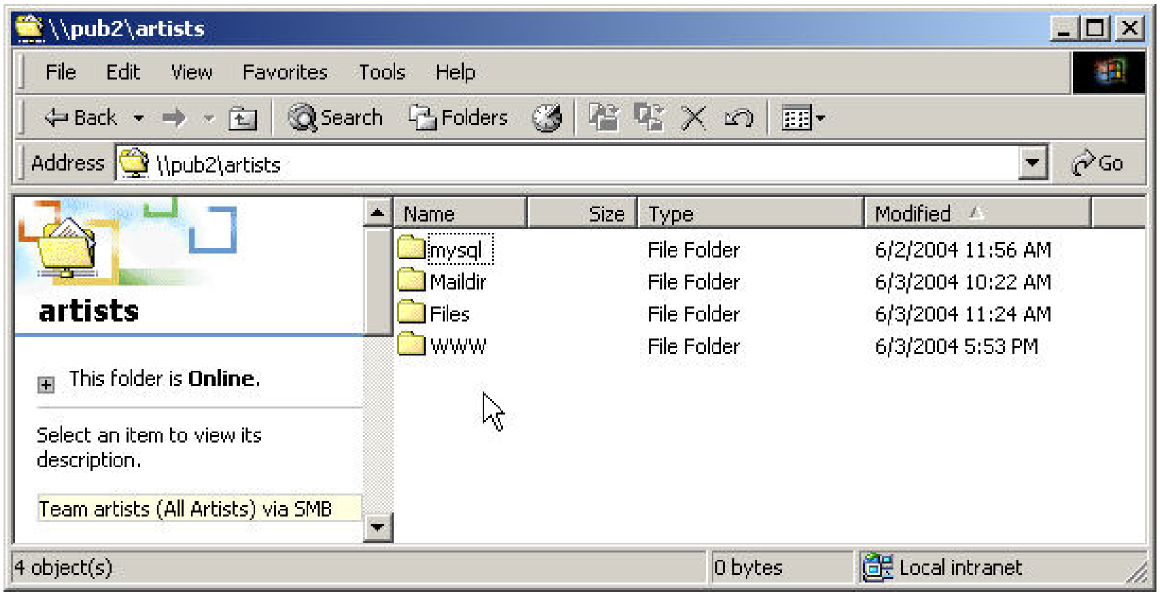
Figure 26 – rsync syntax

rsync syntax:

`rsync` and `–zav` were explained earlier.

1. `--progress` - displays a file progress output, the –P option is actually preferred here ;-)
2. `/home/artists/*` - is the local directory to push out to PUB2
3. `artists@pub2` - artists is the team name at pub2 (where pub2 should be the IP address or the Fully Qualified Domain Name of the target computer) The password prompt following the rsync line is for this account.
4. `::` - a double colon in the destination field tells rsync to copy from the local server to the remote server. The double colon also separates the host name from the path that follows
5. `artists` - is the destination folder or path
6. / - The "/" appended to the trailing directory eliminates any confusion rsync might have with the command. Without it, you might end up with /artists/dir/dir/ or something.

The telnet session shown shows the file synchronization list and reports that there are 157 files to check for synchronization. As this is the first time that rsync has run all 157 files will be transferred. Let's look at the "artists" folder on PUB2 to see if the files were actually copied over.



Address: \\pub2\artists

artists

This folder is **Online**.

Select an item to view its description.

Team artists (All Artists) via SMB

Name	Size	Type	Modified
mysql		File Folder	6/2/2004 11:56 AM
Maildir		File Folder	6/3/2004 10:22 AM
Files		File Folder	6/3/2004 11:24 AM
WWW		File Folder	6/3/2004 5:53 PM

4 object(s) | 0 bytes | Local intranet

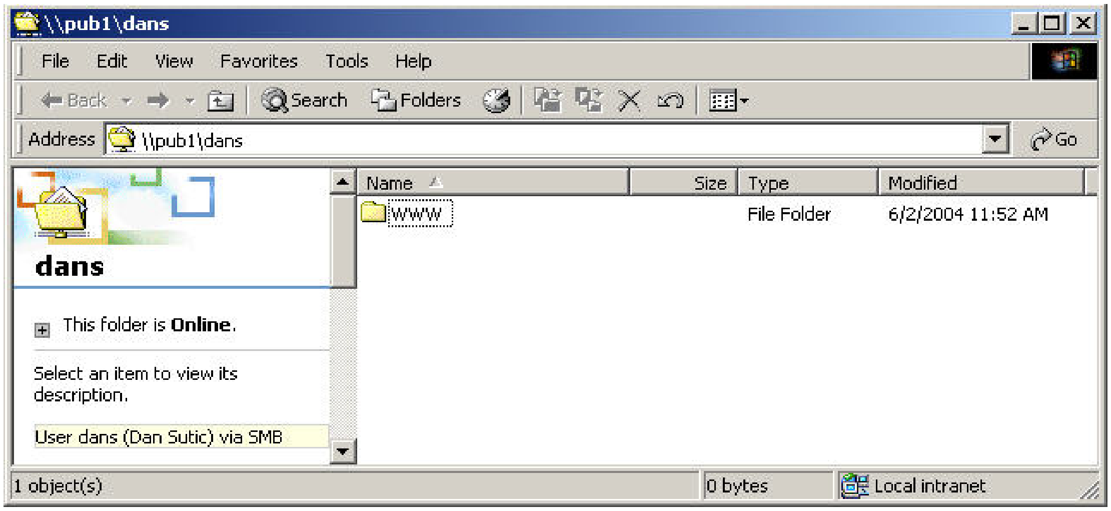
Figure 27 – Files synced in artists folder on pub2

The files are in the \\pub2\artists folder indicating that the rsync command executed correctly.

Pulling:

In this demonstration we will pull the files in the /home/artists/Files folder on PUB2 to the /home/dans/Files/ folder on PUB1.

The following screen capture confirms that the “dans” folder on pub1



Address: \\pub1\dans

dans

This folder is **Online**.

Select an item to view its description.

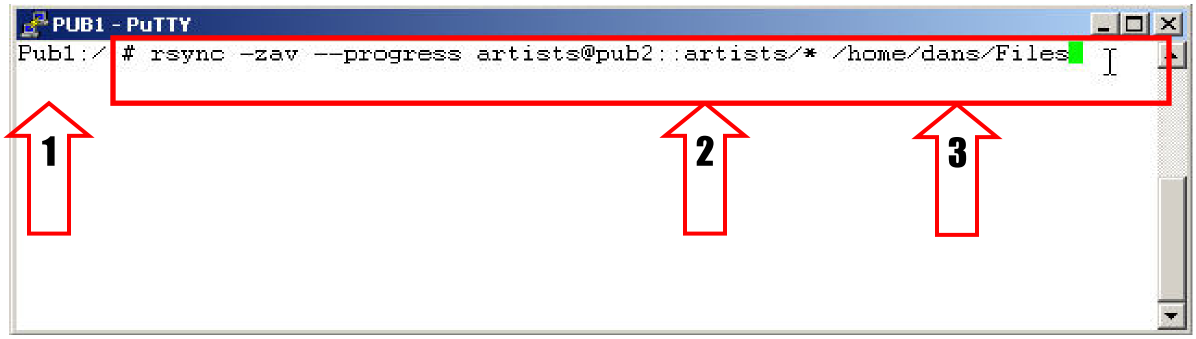
User dans (Dan Sutic) via SMB

Name	Size	Type	Modified
WWW		File Folder	6/2/2004 11:52 AM

1 object(s) | 0 bytes | Local intranet

Figure 28 – Empty dans folder on pub1

There is a subtle difference in that in the telnet session at PUB1 we will pull the files from the “artists” folder on PUB2 back to the “dans” folder on PUB1. A telnet session is opened at PUB1 as shown below.



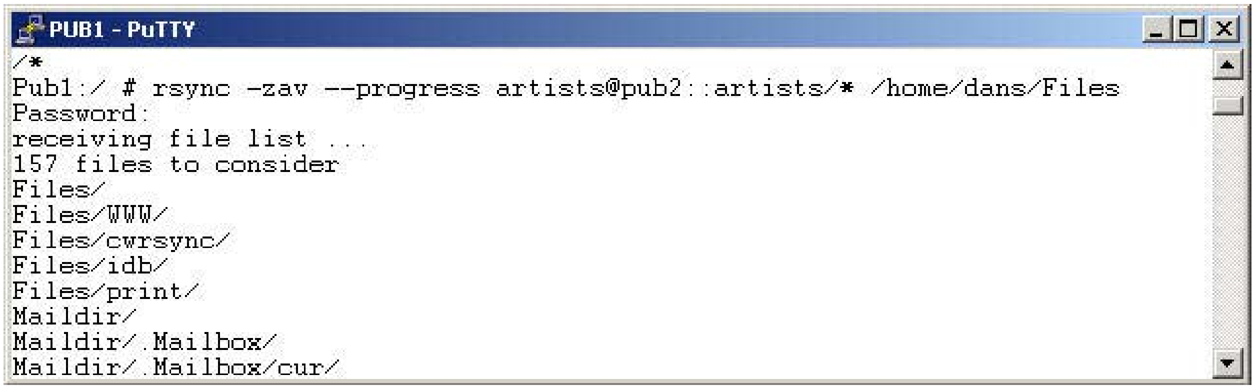
```
Pub1:/ # rsync -zav --progress artists@pub2::artists/* /home/dans/Files
```

Figure 29 – rsync pull syntax

Notes:

1. – The transfer is initiated at Pub1 but the files will be pulled from Pub2
2. - The double colon indicates where the files will be copied from. In this case the transfer will pulled from “artists@pub2” back to the local machine.
3. – “/home/dans/Files” represents the path to the destination folder on the local system.

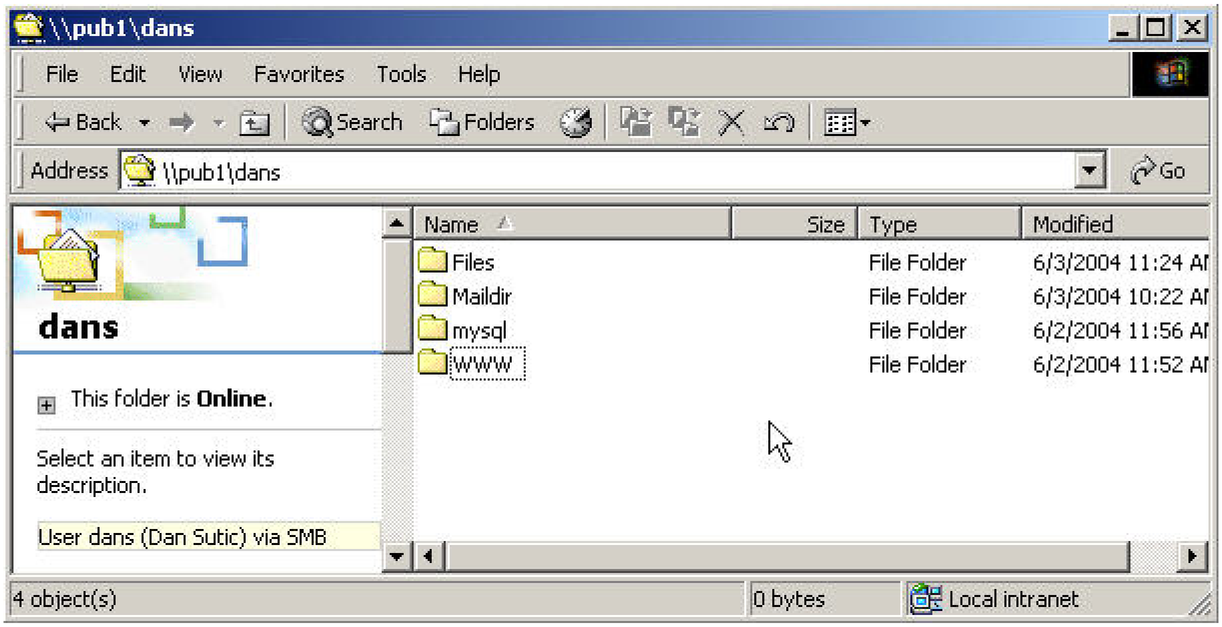
The rsync command is executed and…



```
/*
Pub1:/ # rsync -zav --progress artists@pub2::artists/* /home/dans/Files
Password:
receiving file list ...
157 files to consider
Files/
Files/WWW/
Files/cwrsync/
Files/idb/
Files/print/
Maildir/
Maildir/.Mailbox/
Maildir/.Mailbox/cur/
```

Figure 30 – rsync file progress

The above screen capture displays the files from the “artists” folder that are pulled over to the “dans” folder on pub1 as shown below.



This folder is **Online**.

Select an item to view its description.

User dans (Dan Sutic) via SMB

Name	Size	Type	Modified
Files		File Folder	6/3/2004 11:24 AI
Maildir		File Folder	6/3/2004 10:22 AI
mysql		File Folder	6/2/2004 11:56 AI
WWW		File Folder	6/2/2004 11:52 AI

4 object(s) | 0 bytes | Local intranet

Figure 31 - Files synced in dans folder on pub2

The screen capture above shows that the files have been copied over from the “artists” folder on PUB2.

Automating rsync with cron

Prior to automating rsync between Nitix servers you will need to install and configure the cron utility which is covered in our ***cron How-To***.

Here, we’ll demonstrate how to create a *rsync.sh* script file to automatically run an rsync command on a recurring basis. In this example we’ll configure the *rsync.sh* script to run the rsync command to synchronize files between folders on our two Nitix powered servers and then have cron execute the rsync.sh script at 5 minutes past midnight every Monday through Friday. See our ***rsync How-To*** for a more complete explanation of how rsync works.

The first step here is to get your rsync command tested and working from the command line before entering that command into the script file. In our example we’ll synchronize the */home/artists/Files* folder on a server called PUB1 to the */home/test/Files* folder on the PUB2 server.

The command to do this by pushing the synchronization out from the PUB1 server to the PUB2 server would be:

```
rsync  -zavP home/artists/Files/* cron@PUB2::cron/Files
```

Once you have this command working from the command line you can put the command into a file and then make that file executable. The name of the script file can be anything and in this example we'll call the file rsync.sh.

As you may use script files to automate many tasks it might be a good idea to create a bin folder below the cron home directory to hold all of our executable cron files. To create the bin directory enter:

```
mkdir /home/cron/bin
```

The next step is to create the script file, **rsync.sh,** with the following lines in the new **/home/cron/bin** folder.date

```
#!/bin/sh
RSYNC_PASSWORD=1234         #1234 is the password of the destination team account
export RSYNC_PASSWORD
rsync –zavP home/artists/Files/*  cron@192.168.10.158::cron/Files
```

We'll then make the file executable by entering:

```
chmod  770  /home/cron/bin/rsync.sh
```

Finally, you will need to enter your script file into the **_root** file to have it executed automatically by the cron service.

Open the /home/cron/Files/crontabs/_root file and enter the following command

```
#  run rsync.sh at 5 minutes past midnight Mon thru Fri
5 0 *  * 1-5     /home/cron/bin/rsync.sh
```

Save the file and you're done.

An Important Note on File Permissions:

When you rsync files from one server to another, this can be any Windows or *NIX server, the transferred files inherit the permissions of the destination folder. This means that user1 on srv1 will still be able to access her files if they have been transferred with rsync to a user1 folder on SRV2. However, if the files are copied to any other user or team account folder the user will not have access to the files unless the user is a member of the team folder where the files were copied to. Note however that all other members of the team would also have access to the files.

Summary

In this How-To we learned what rsync does, how to configure rsync from a windows client using cwrsync and how to run rsync, inside a Batch file, automatically as a windows Scheduled Task. We then examined two ways, a push or a pull, to synchronize files between Nitix servers. Next we learned how to automate the recurrence of rsync jobs with the use of the cron utility. We also noted that files transferred with rsync lose the source permissions and inherit the permissions of the destination folder.

About Net Integration Technologies

Net Integration Technologies Inc. (NITI) is a software developer that delivers autonomic, Linux-based server operating system (OS) solutions to the SMB market. The company's revolutionary Nitix server OS sets new standards in stability, security, affordability and ease-of-use for small to mid-sized businesses. Nitix-powered servers have earned the company numerous awards and rave reviews from notable publications like PC Magazine, CRN, eWeek and InfoWorld. Established in 1997, the company has built a worldwide Approved Partnership Program with over 1,500 reselling partners. NITI is headquartered in Markham, Canada with additional offices in Montreal, the United States and Europe. For more information, please visit: http://www.nitix.com.