

Rsync Internet Backup Whitepaper



WHITEPAPER

BackupAssist Version 6

www.BackupAssist.com

Cortex I.T.

© Cortex I.T. Labs 2001-2008

Contents

Introduction	3
Important notice about terminology.....	3
Rsync: An overview	3
Performance	4
Summary	5
Best practices and FAQ.....	5
“Cutting to the chase” – use these guidelines for maximum success	5
How does Rsync perform on files and directories?.....	6
Scenario 1: File system with 50,000 files, 50 GB total; 50 files of total size 50 MB have changed.....	6
Scenario 2: The file system is backed up via NTBackup, which results in a 50GB bkf file.	6
Does Rsync preserve file attributes with the backup?	6
Is there a maximum size or number of files in my data set?	7
How many simultaneous backups can I run to my Rsync destination?	7
Can I backup Exchange databases, SQL databases using Rsync?.....	7
Does BackupAssist compress and encrypt my data?	7
Can I use Rsync to synchronize my drive images offsite?.....	8
Are there any caveats to using a dedicated NAS device as my data host?	8
Rsync Data Hosts.....	9
Daemon mode vs. Rsync over SSH.....	9
Using a Windows Rsync Data Host.....	9
Setting up a Windows Machine to act as an Rsync Server	9
Prerequisites:	9
Installing cwRsync:	10
Installing CopSSH:	10
Activating a user.....	10
Configuring the BackupAssist client for a Windows server.....	12
Using a Linux Rsync Data Host.....	13
Creating logons on your data host	13
Configuring the BackupAssist client for a Linux server	14
Setting up a NAS Rsync Server	15
Rsync Server Data Seeding	15
Option 1 – bringing your data host onsite to perform the seed	15
Option 2 – seeding a permanently offsite data host.....	16
Troubleshooting and Support.....	17
Appendix	17

Troubleshooting.....17

Introduction

BackupAssist provides a simple and automated solution for organizations who want to store a backup copy of their data offsite via LAN or WAN using an efficient and effective transfer method.

This whitepaper outlines:

- how the Rsync client works
- performance and best practices
- how to setup Windows and Linux machines to act as your data host
- how to use Rsync-enabled NAS devices as your data host for a turnkey solution.

Important notice about terminology

In order to avoid confusion about the use of the words “client”, “server”, “Windows Server”, “Rsync Server”, and so on, we will use the following terms to avoid ambiguity:

Data Host – the remote machine on which you store your data.

Rsync Server – the same as the data host – specifically referring to the machine running Rsync that accepts incoming connections and data from Rsync clients

Rsync Client – a machine that contains your working data (typically a file server) that has BackupAssist installed. BackupAssist comes packaged with the Rsync libraries necessary to transfer data to the Rsync Server during a backup.

Rsync: An overview

Rsync is an open source software application, originally written for Unix systems, but now also running on Windows and Mac platforms. It is used to synchronise files and directories from one location to another while minimizing data transfer between each location.

The data transfer is minimised using an algorithm that will transmit, roughly speaking, only the parts of the backup selection that have changed, right down to the bit level. (This technology is also known as in-file delta incremental transfer.) Along with this minimized data transfer Rsync also compresses all data packets sent, further reducing transfer overheads.

Rsync uses a checksum method to perform this bit level data transfer. This method creates a short alphanumeric string based on the data it represents. Rsync first checks whether any data has changed by looking at the file size and modification date. If no data has changed, Rsync will not transfer any data, saving time and bandwidth. If files do not match, Rsync uses a checksum method called a ‘rolling checksum’ on the changed files to see where it has been altered or appended. It will then transfer only the altered or appended data within the file. Rsync can cater for inserted or added data, removed data as well as shifted data, with a minimum transfer overhead.

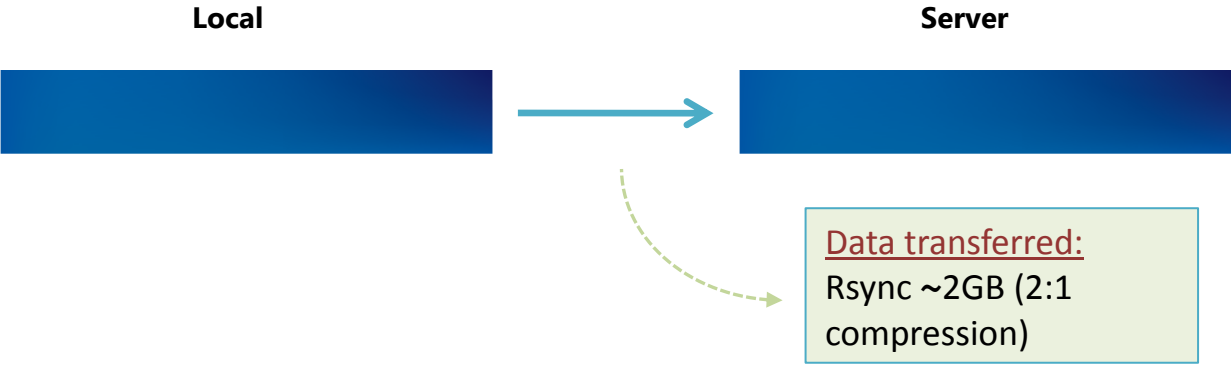
In real terms, that means **more efficient use of your bandwidth and data allowances**. As Rsync will only transfer data that has changed and knows when file alterations or movements have occurred, your Internet based backups will take a lot less time when compared other methods such as FTP.

Performance

To help better understand how Rsync transfers work we will take a look at a hypothetical three day backup scenario.

Day 1:

We begin with a data file of 4GB backed up using three different methods; Rsync, FTP and Incremental drive imaging.

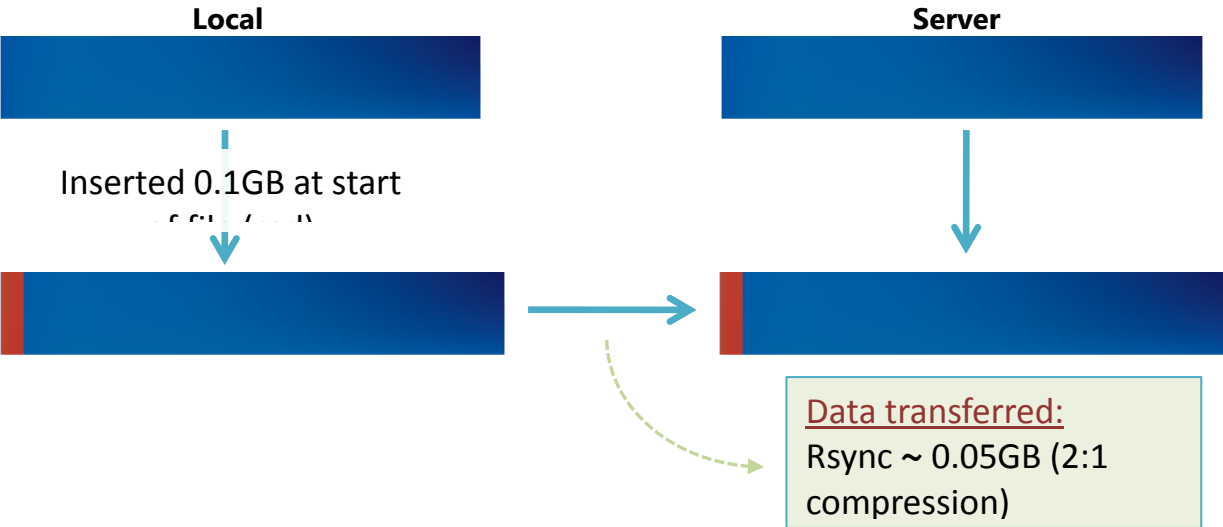


Looking at this first backup we see that for the initial data transfer there is a 100% transfer for both Incremental drive imaging and for FTP; thanks to Rsync's packet compression we see a 50% reduction in the initial transfer.

Note: depending on your Rsync server's setup this initial overhead can be removed by seeding your backup server locally, a method we will discuss later in this paper.

Day 2:

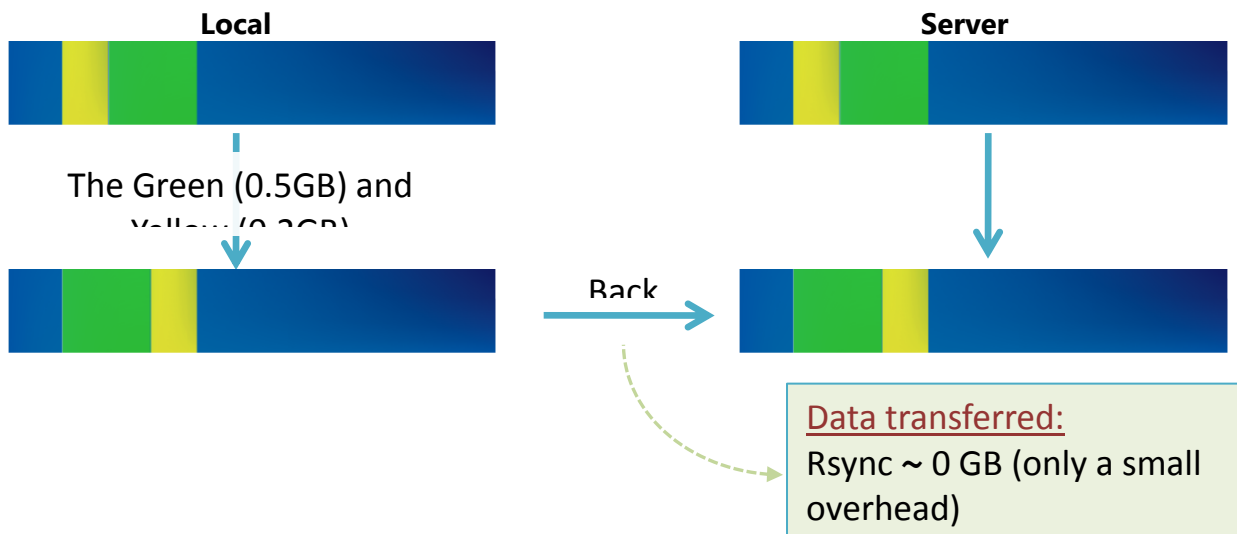
On the second day we have added a further 0.1 GB to the start our data file.



We can see that both FTP and Incremental drive imaging perform a full backup of the file. Rsync however, only backs up the changed data within the file, and compresses the sent data, resulting in a 50mb transfer.

Day 3:

This day no data has been added, but data *has* been shifted within the file.



Rsync is able to recognise that this data is already on the backup server and will reorganise the file with a minimal instruction file. Incremental drive imaging is also aware that the data was moved; however it must re-back-up the moved data as this section does not match the data source. FTP once again has to do a full backup of the source data.

Summary

As demonstrated in this example, Rsync delivers substantial performance gains. With the ability to check what data is still the same, then append, remove or modify it as necessary to match the local source it can greatly reduce backup overhead.

The key benefits of Rsync:

- Improves offsite backup speed through bandwidth optimization.
- Reduces network data transfer by transferring only new data
- Open standard protocol – for maximum compatibility and flexibility in choosing a backup destination.

Best practices and FAQ

“Cutting to the chase” – use these guidelines for maximum success

- Use Rsync to back up data straight from the file system. This will make sure that the data is in the smallest data blocks, resulting in the fastest possible backup. You will find this preferable to using Rsync on a backup or image of the file system.
- When your job is first set up, you should “seed” your data on the data host by using a USB HDD to physically transport the data, or if using a NAS device, running the job once over a local network. Specific instructions on backup seeding can be found later in this document.
- Run your Rsync job regularly. Regular daily interval backups will ensure that you keep your data transfer to a minimum as well as keeping a safe, secure up-to-date backup.
- For maximum protection, use your Rsync backup as part of your complete backup plan. Use Rsync to back up your critical data offsite, along with a drive image, as well as conventional, local, archive file backups.

The following FAQs explain how we devised these guidelines and explain in more detail why we make these recommendations.

How does Rsync perform on files and directories?

Rsync performs best working directly on the file system, backing up normal files and directories. Rsync does not perform nearly as well synchronizing backup files offsite.

Let's look at example to see why that's the case.

Scenario 1: File system with 50,000 files, 50 GB total; 50 files of total size 50 MB have changed.

Rsync is able to identify which of the 50 files have changed, and for those files, it determines the in-file deltas. It calculates checksums on 50MB of data, and can complete the backup in a matter of minutes. The amount of data transferred will be around 20MB for typical documents.

Scenario 2: The file system is backed up via NTBackup, which results in a 50GB bkf file.

Rsync will detect that the single bkf file has changed, and needs to determine the in-file deltas. It needs to calculate checksums on 50GB of data, which may take hours. Additionally, we have found that even if the underlying file system changes very little, about 10% of a bkf file changes from day to day and needs to be transferred. So, about 5GB will be transferred.

We see here that it is greatly preferable in terms of bandwidth and CPU time that Rsync operates on the underlying file system rather than a backup of that file system.

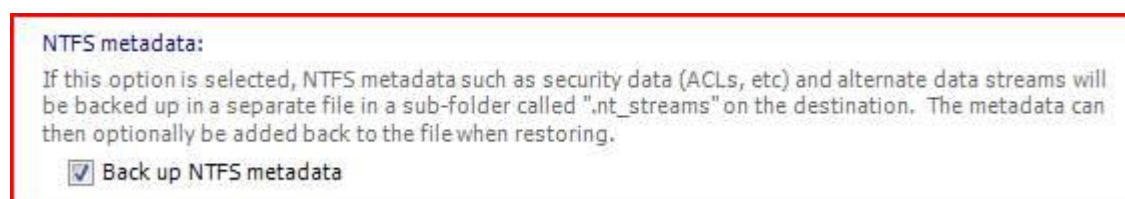
Does Rsync preserve file attributes with the backup?

Because Rsync works on top of the Cygwin Unix emulation layer, it does not recognize Windows file attributes (e.g. readonly, hidden, system, etc) or NTFS security attributes (i.e. access control lists). NTFS alternate data streams are also not supported, and as Unix does not have a concept of file creation time, this is also not preserved. The following file system attributes will be preserved at the backup destination when using Rsync to transfer data:

File attributes at destination	Preserved?
Windows File Attributes	✗
Creation time	✗
Last access time	✗
Last modified time	✓
NTFS security (ACLs)	✗
NTFS alternate data streams (ADSs)	✗

“*” The ACLs and ADSs are preserved if destination is NTFS (not Linux or ReV)







There is, however, an option in BackupAssist, within the **Rsync options** tab, that allows you to have NTFS metadata stored on the backup destination as well:



This will be checked by default for new jobs created in BackupAssist v6. If enabled, NTFS streams, such as alternate data streams and security data will be saved to a separate file on the destination

and then added back to the file as part of the restore process when using the BackupAssist Restore Console. So while these attributes are not "preserved" on the files backed up to your Rsync destination, they will still be restored.

The table below outlines what file system attributes are preserved at the backup destination when the NTFS metadata option is enabled:

File attributes at destination	Preserved?
Windows File Attributes	
Creation time	
Last access time	
Last modified time	
NTFS security (ACLs)	
NTFS alternate data streams (ADSs)	

Is there a maximum size or number of files in my data set?

In theory, there's no limit to the number of files or directories that you can Rsync – apart from the practical limitation of RAM.

Even though Rsync only transfers the data that has changed on a day to day basis, it still must read all of the data in the file set to check which data has changed. This makes Rsync internet backups a disk/CPU intensive operation that can take longer and longer the more your data grows, no matter how little data has actually changed. We recommend that wherever possible, you use one of the other backup methods provided in BackupAssist (such as the BackupAssist Zip Engine) to regularly archive infrequently used data, so the amount of actual data in day to day use is minimized.

We have run tests on several different file systems – a typical file system of 70,000 files and 24 GB with fewer than 50 MB of daily changes can be synced in around 10 minutes. The largest file system we've tested is of 200,000 files and 100 GB, which took 20 minutes to sync minimal changes.

How many simultaneous backups can I run to my Rsync destination?

With Rsync, simultaneous connections may become unreliable with heavy data transfer loads, and it is therefore recommended that you limit connections to your own Server to five at any one time. Depending on data storage requirements and the bandwidth speeds available, you may increase this number with caution.

Can I backup Exchange databases, SQL databases using Rsync?

Yes. The BackupAssist Rsync engine includes fully integrated support for VSS application backup and restore. Microsoft applications such as Exchange Server, SQL Server, SharePoint and Hyper-V are all fully supported, as well as any other VSS-aware application that uses standard VSS restore methods.

Simply choose the VSS application that you want to back up from the list of detected applications in the **Files and folders** tab. You can even drill down and choose individual components (databases, storage groups, etc) to backup. Application restore is just as easy using the BackupAssist Restore Console: browse the contents of a backup and select the application(s) you want to restore.

Does BackupAssist compress and encrypt my data?

BackupAssist supports encryption and compression on the server, plus full NTFS streams support, for a complete solution for remote backup.

Adding to the super bandwidth-efficient algorithm that Rsync provides, BackupAssist for Rsync offers industry standard encryption for data stored on the Data Host. This means that your data is safe "in the cloud", making external hosting a safe and secure option.

Your files are also automatically compressed on the Data Host, which reduces the amount of disk space used on your hosting company. BackupAssist for Rsync utilizes four distinct types of compression:

- 1) Effective transfer compression via only sending changed data
- 2) All data packets are compressed and encrypted during transfer
- 3) Single Instance Store (SIS) uses hard link technology to prevent storing the same files more than once across backups on your Host.
- 4) The source data is encrypted and compressed in a rsync-friendly way before transmission, effectively minimizing the space used by files on the server even further

Note: if you enable or disable encryption for an Rsync job, BackupAssist will need to "re-seed" the backup to the Host with a full set of data (i.e. the next backup will be a full backup regardless of how many files have changed).

Can I use Rsync to synchronize my drive images offsite?

We recommend that you select the underlying files for an Rsync backup rather than an image backup of your file system.

However, that said, drive images are more suitable for Rsync than many other types of backup, provided they are uncompressed and unencrypted, but the checksum process will be CPU intensive. We have found that on typical servers checksums can be performed at a rate of about 100-120GB per hour, during which time the server's CPU is at about 30% on a single core. [Note: on multi-core processors, this means that CPU usage is quite low.]

The time to backup via Rsync can be approximately calculated as:

$$2 * \text{checksum time (one checksum for each end)} + \text{network time}$$

So if you really, really want to do it, you can, but we believe there are better ways.

Remember - the purpose of doing multiple backups is redundancy. That means protecting your data in different ways, to different locations. If you synchronize a drive image offsite, you run the risk that the drive image is bad and you have just lost all of your backup data. Instead, if you back up your underlying file system using Rsync, and your image is bad, you still have the files and folders at your remote site.

The use of Rsync as a backup solution is best suited to a regular file system. Due to the creation of rolling checksums on altered backup files, it is disadvantageous to have files combined into an archive. This is because only files that are flagged as altered will have the rolling checksum performed on them.

If you have a very large single archive file (>100 GB) it will take much longer to complete the rolling checksum process, even if only a small element has changed. This may or may not be a problem, depending on the processing power of your Rsync server.

Are there any caveats to using a dedicated NAS device as my data host?

Many dedicated NAS devices offer built-in support for Rsync. While this can be convenient to set up, many of these devices use low-powered processors which can result in a performance hit if you are backing up large files (several GB or larger in a single file). The following example illustrates the difference in backup time for a dedicated QNAP NAS device, versus an ordinary desktop Linux machine. The initial backup is a single 18.8GB file. The second backup consists of about 200MB or changes to that file.

Device	QNAP TS-209II with rsync 2.6.6	Ubuntu 9.04 desktop with rsync 3.0.5
Initial backup	7 hours 55 minutes	1 hours 22 minutes

Second backup	4 hours 57 minutes	0 hours 35 minutes
----------------------	--------------------	--------------------

Rsync Data Hosts

As rsync is an open protocol, you have several options for storing your data.

They can be broadly summarized as follows:

1. Any third-party data center or cloud provider that supports rsync
2. Do-it-yourself – any rsync server – such as an rsync-enabled NAS device, Windows, Unix or Linux machine. *For example, multiple servers may be located in different branch office locations, your VAR's office, etc.*

Third-party data centers have the advantage of high availability networks and some datacenters also offer geo-redundant storage. At the time of writing (November 2011), we provide a specialized destination to support backup to Amazon S3 via the s3rsync.com service.

See <http://www.backupassist.com/blog/support/using-backupassist-for-rsync-with-amazon-s3/> for information on setting this up. We hope to add support for other data centers in the future.

The do-it-yourself approach has the advantage of keeping data in your control, and a lack of monthly hosting fees or limits to the amount of data backed up. Using your existing internet connection and hardware can be a cost effective solution.

A popular choice of destination is an rsync-enabled NAS device placed in the business owner's home. Legal firms especially appreciate this approach, since control over information is their primary concern.

The following chapters describe the DIY approach.

Daemon mode vs. Rsync over SSH

Rsync servers can be one of two flavors:

- Rsync over SSH (preferred) – this runs rsync via a secure shell (SSH, port 22) which means that all traffic over the internet is encrypted. User access control is modified by editing user accounts on the server.
- Daemon mode – this runs rsync as a normal TCP/IP service. User access control is modified by editing the rsync.conf file. Internet traffic is not encrypted.

In the following chapters, the Windows and Linux data hosts support rsync over SSH. However, some NAS devices do not, and daemon mode must be used instead. Daemon mode is still an acceptable solution provided a secured LAN/WAN (such as site-to-site VPN) is used.

Using a Windows Rsync Data Host

Setting up a Windows Machine to Act as an Rsync Server

To use rsync with SSH you will first need to install both SSH and rsync on your Windows server. We suggest the use of CopSSH and cwRsyncServer. An installer for these packages can be issued from us by browsing [here](#) and filling out the web form.

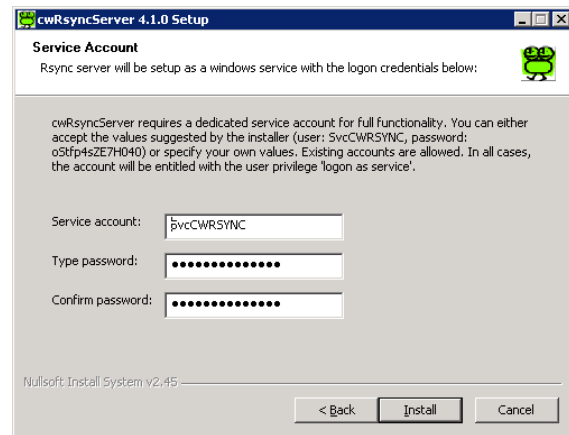
Prerequisites:

- A Windows 2000 or later machine with network connectivity and space to store backup data.
- The cwRsyncServer installer.

- The CopSSH installer.
- BackupAssist v5.1.0 or later installed on the Windows machine you want to back up (i.e. the client).

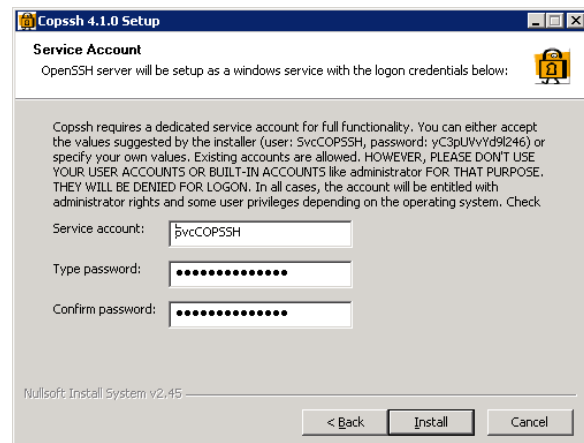
Installing cwRsync:

- 1) Run the cwRsyncServer installer.
- 2) Continue through the installation wizard, installing the package to any location you choose.
- 3) During the installation you will be presented with the popup on the right. We suggest leaving the SvcCWRSYNC account as is. Write down the password provided.
- 4) Click 'Install' to install the package. Once this is completed cwRsync should be present on your system.



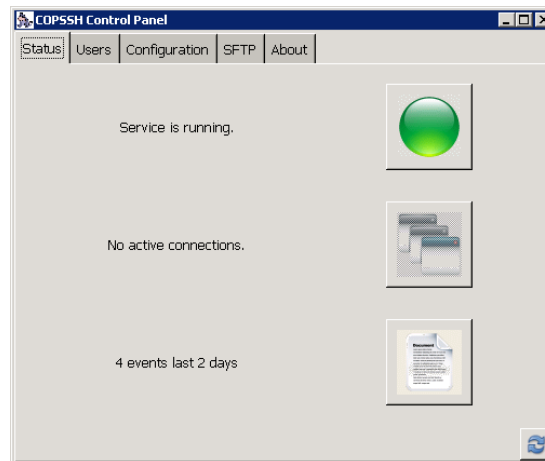
Installing CopSSH:

- 1) Run the CopSSH installer.
- 2) Continue through the installation wizard, installing the package to any location you choose.
- 3) During the installation you will be presented with the popup on the right. We suggest leaving the SvcCWRSYNC account as is. Write down the password provided.
- 4) Click 'Install' to complete the process of installing CopSSH on your system.
- 5) Later in the installation you will be presented with the below popup. At any time after the installation you can access 'Activate a user' from your start menu to allow SSH access to that user. You must activate at least one user before you will be able to register an rsync client. Click 'OK' to continue your installation.



Activating a user

If you are planning to use SSH, then before you register a BackupAssist client with your rsync server, you must activate a user with CopSSH. In the start menu, under All Programs -> CopSSH, select 1. CopSSH Control Panel. The screen below will appear:



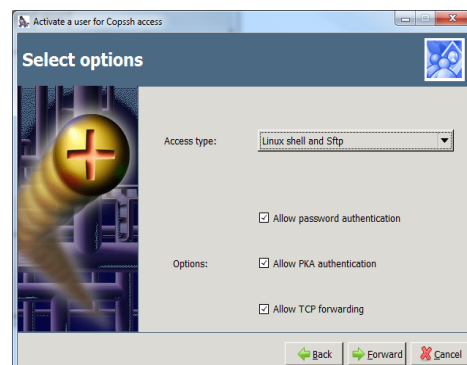
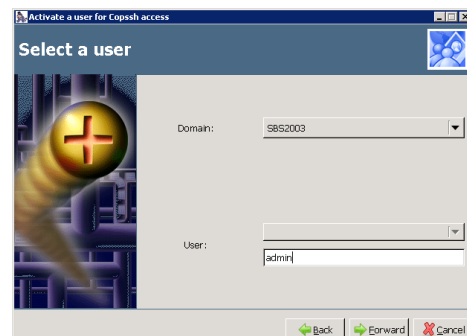
To start the process to activate a user, click on the Users tab across the top of the user interface. The following screen will appear. Click on the 'Add' button to bring up the wizard to activate a user.

DO NOT ACTIVATE YOUR ADMINISTRATOR ACCOUNT

Doing so will cause a lock down on the account due to CopSSH's security settings. We recommend activating a newly created account.

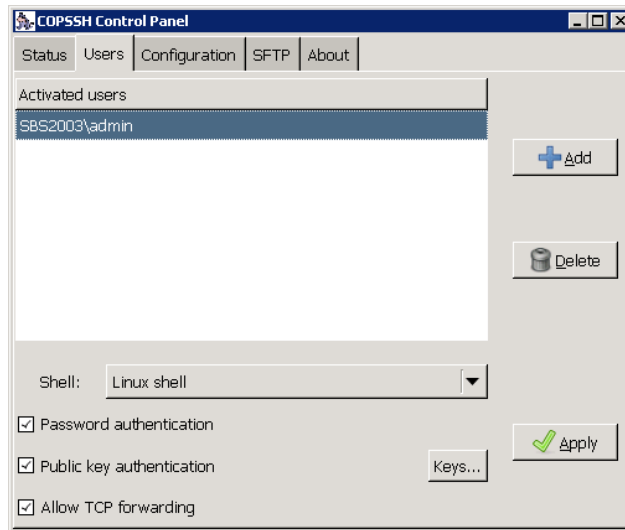
During the 'Activate a User' Wizard:

- 1) Click 'Forward' on the opening screen.
- 2) On the second screen, select the Domain and type in the user which you wish to activate. Click Forward once complete (admin is a manually created account we'll use for this example).
- 3) Change the 'Access Type' to Linux Shell and Sftp using the drop down menu. Leave all 'Options' enabled (as they are by default):



- 4) On the fourth screen, click on 'Apply' to complete the wizard and activate the user.

The user should now be showing as activated within the CopSSH Control Panel:



Your user's home directory will be located at (for example) `C:\Program Files\ICW\home\justin`. The location of this directory can be changed by editing the file `C:\Program Files\ICW\etc\passwd`.

Note: If you need to uninstall the cwRsync server at all, please be aware the two Windows service users 'SvcCOPSSH' and 'SvcCWRSYNC' are not removed. If you then re-install the cwRsync server package the Windows users cannot be recreated and the passwords do not match. This ultimately means the CopSSH and rsync services will not start on the server.

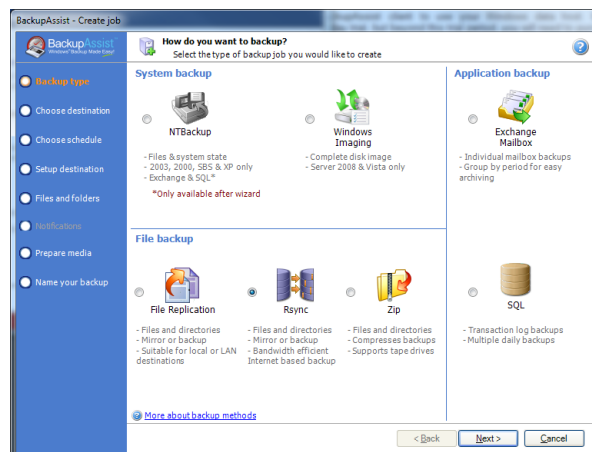
The fix is to uninstall and remove the users manually then re-install to add the users again with known passwords.

Configuring the BackupAssist client for a Windows server

Now you should configure the BackupAssist client to use your Windows data host. Install BackupAssist. You will have a free 30 day trial, but beyond this trial period, you will need to purchase a license for "BackupAssist for Rsync" to continue using it.

To begin this process, create a new BackupAssist backup job.

1. Launch the BackupAssist console and choose File > New backup job from the drop-down menu.
2. Select Rsync from job type choices and then click 'Next'. (see screenshot right)
3. Select **Rsync** as your server type and click 'Next'.
4. Choose a Schedule and click 'Next'.



5. In the Rsync Server options section:

- i. Enter your Rsync server name (or IP address), and choose “Rsync over SSH”. This option ensures that your data is encrypted during transmission.
- ii. Under “Path on server”, type in the path to your backup directory.

Note: It is best to use a new, empty directory for this path. The parent directory must exist though the sub directory will be created when the job is first run, e.g. /parent/sub_directory/. If your data host is running Windows, you can enter a normal Windows path here, such as “C:\Backups”. You can also enter a path relative to the user’s home directory by starting with a tilde (“~”), e.g. “~/Backups”.

- iii. Enter a username that was activated while setting up your Rsync host
- iv. Click ‘Register with server...’. You will be prompted to enter the user’s password, then BackupAssist will create a public/private key pair to authenticate you to the data host. This will be the only time you need to enter your password. If successful, a message will appear to the right of the button.
- v. Click the ‘Test connection...’ button to test communication with the Rsync server. If this step fails but the registration succeeded it’s probably that the ‘Path on server’ cannot be accessed. Try ‘~/Backups’.

Using a Linux Rsync Data Host

Most FreeBSD and Linux servers can be used to host backup data. BackupAssist has two requirements – that the data host has an SSH server and Rsync installed.

All major Linux distributions (such as Fedora, RedHat Enterprise, Ubuntu, Debian) have these two prerequisites available as install options. The most common SSH server is OpenSSH.

Side note: You can choose to run Rsync as a daemon on your Linux server. (However for security reasons, we do not recommend this – use Rsync over SSH instead.) If you choose to run Rsync in daemon mode, you will not need to have the SSH service installed. For instructions on setting up BackupAssist to connect to an Rsync daemon please view the “Configuring the BackupAssist client for a NAS server” section of this whitepaper.

To determine if your system has these prerequisites installed, log into your system and start a shell. Then type:

man rsync – this should return the man page for Rsync if installed. Type ‘q’ to exit the man page.

man sshd – this should return the man page for sshd if installed. Type ‘q’ to exit the man page.

If not installed, you should use your distribution’s software package manager to install these packages. Most commonly they can be found under the “Server” or “Security” categories.

Creating logons on your data host

The next step is to create logons on your data host. We recommend creating a separate logon for each client. For example, if you host data for 5 different companies, create 5 different accounts so that each company will only be able to see their own data.

You should also make sure that each client’s home directories are on a partition that contains sufficient space to host their data.

You **must** also change the permissions on each user’s home directory, or else most SSH daemons will not allow you to connect to the server using the public/private key method (which BackupAssist

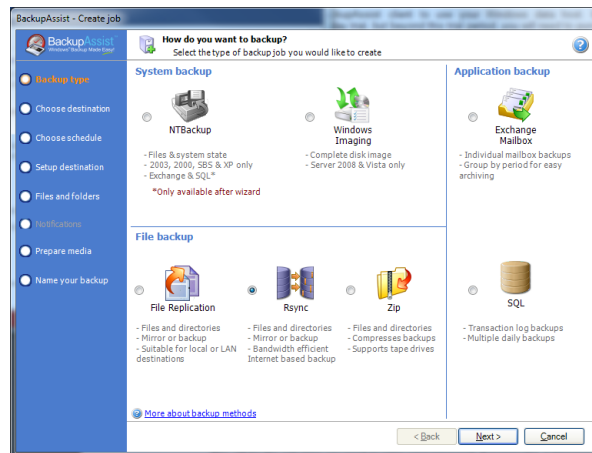
uses). To do this, use the chmod command – for example for a user “fred”, type in the following (when logged on as root): chmod 700 /home/fred

Configuring the BackupAssist client for a Linux server

Now you should configure the BackupAssist client to use your Linux data host. Install BackupAssist. You will have a free 30 day trial, but beyond this trial period, you will need to purchase a licence for “BackupAssist for Rsync” to continue using it.

To begin this process, create a new BackupAssist backup job.

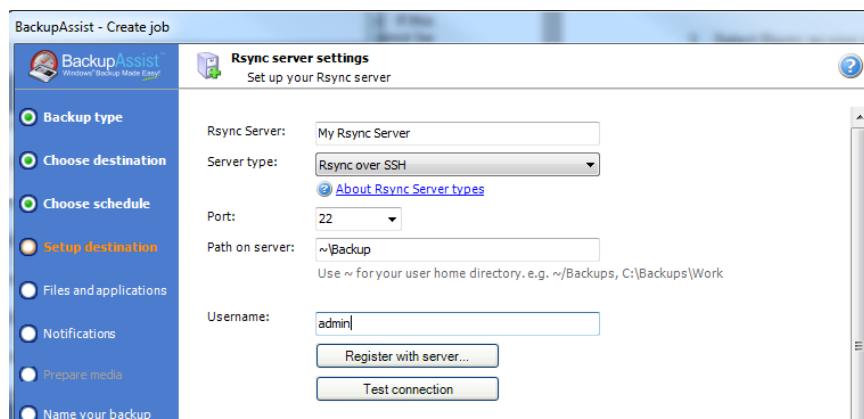
1. Launch the BackupAssist console and choose File > New backup job from the drop-down menu.
2. Select Rsync from job type choices and then click ‘Next’. (see screenshot right)
3. Select Rsync as your server type and click ‘Next’.
4. Choose a Schedule and click ‘Next’.



5. Now, in the Rsync Server options section (see the screenshot below)
 - i. Enter your Rsync server name (or IP address), and choose “Rsync over SSH”. This option ensures that your data is encrypted over the wire.
 - ii. Under “Path on server”, type in the path to your backup directory.

Note: It is best to use a new, empty directory for this path. The parent directory must exist though the sub directory will be created when the job is first run, e.g. /parent/sub_directory/. You can use either an absolute path by starting with a slash (e.g. “/var/Backups”), or a path relative to the user’s home directory by starting with a tilde (e.g. “~/Backups”).

- iii. Enter your Username (the logon that you created in step ii immediately above)
- iv. Click ‘Register with server...’. You will be prompted to enter in your password, and then BackupAssist will create a public/private key pair to authenticate you to the data host. This will be the only time you need to enter your password. If successful, a message will appear to the right of the button.
- v. Click the “Test connection...” button to test communication with the Rsync server.



Setting up a NAS Rsync Server

Backing up to an Rsync-enabled NAS can be a very effective solution. The advantage of using a NAS is that as an appliance, it can be close to a turnkey solution and easier to manage.

Each NAS is different, and some support Rsync over SSH, whereas others only support Rsync Daemon mode. There is however a list of requirements that must be met in order for BackupAssist to connect to the device.

To use your NAS as an Rsync data host you will need:

- A NAS that is running Rsync as a daemon, or one that has Rsync and an SSH service running.
- Setup a share to act as a root directory for your Rsync backups and allow read and write permissions to this directory.
- If your NAS requires a password to connect to the Rsync service you will need for BackupAssist to authenticate to it.
- Your NAS will need to have the correct ports open for your Rsync Daemon or SSH service (873 and 22 respectively)

Many of these options vary from device to device, so you will need to consult your manual to correctly setup the destination.

Recommended reading: If you are looking for a NAS device to use as an Rsync server please read our [Rsync NAS Hardware Compatibility List \(HCL\)](#) for a list of NAS devices that have been tested with BackupAssist. The HCL contains **step-by-step setup instructions** for the compatible devices.

Rsync Server Data Seeding

Rsync backups are in-file delta incremental backups. The first time you perform your backup, no data will exist on your data host, so a full backup is required. Similarly, if you enable or disable encryption for an Rsync job, BackupAssist will need to "re-seed" the backup to the Rsync backup destination with a full set of data (i.e. the next backup will be a full backup regardless of how many files have changed).

Seeding your backup via a slow Internet connection may not be practical, so two methods are provided here to seed your data host. Once the initial seed to the data host is complete, each successive backup will be an incremental backup of only data that has changed.

Option 1 – bringing your data host onsite to perform the seed

This method is suitable for "standalone" data hosts (where a data host is not shared among multiple clients) that can be physically transported onsite – such as NAS devices.

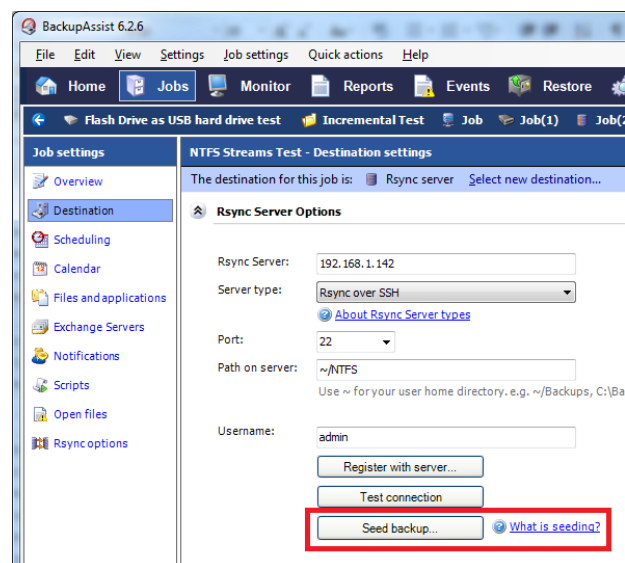
Seeding your data is easy – simply follow these instructions:

1. Connect your data host to the LAN, and make a note of its IP address / Hostname
2. Create your BackupAssist Rsync job and run it at convenient time and wait for it to complete.
3. Move your NAS to its permanent location
4. Update the job settings in BackupAssist to reflect the new IP address / Hostname

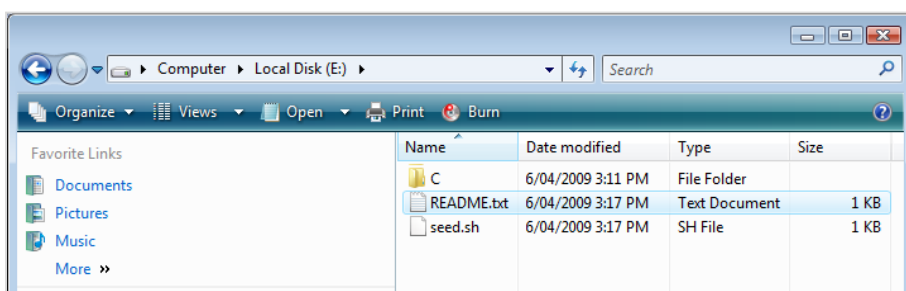
Option 2 – seeding a permanently offsite data host

You can use BackupAssist to automatically seed data offsite using a USB HDD or an equivalent portable disk-based device, which can be then physically transported to the data host and the data uploaded locally from there. Seeding your data using this method is simple:

1. Connect a USB HDD or an equivalent portable disk-based device to the machine running BackupAssist.
2. Create your BackupAssist Rsync job and access the **Destination** window by clicking **Edit** from the top menu > selecting your Rsync job > and clicking **Destination** from the left menu:



3. Click the **Seed backup** button and select the location of an empty folder on your USB HDD.
4. Once the seed is complete, your USB HDD destination should contain: 1) a **directory** with the seeded data; 2) a **README.txt** file containing instructions on how to copy the seed to your Rsync server; and 3) a **.sh script file**, which is used to copy your data to your Rsync server:



5. Transport the USB HDD containing the seed to the site where your Rsync server is located and connect the device to this machine.
6. To copy the seed to your Rsync server:

Note: these instructions are contained in the README.txt file located on your USB HDD

For a Windows server (assuming the seed is located on **E:\SeedFolder**)

- i. Go to Start menu > CopSSH > Start a Unix BASH shell
- ii. Enter the following command:


```
bash "/cygdrive/e/SeedFolder/seed.sh"
```

For a Linux or Unix server (assuming the seed is located in `/mnt/usbdrive/SeedFolder`)

- i. Run the following command in your shell:

```
bash "/mnt/usbdrive/SeedFolder/seed.sh"
```

7. A complete seed of your data should now be copied to your Rsync server. Each successive backup from now on will be an in-file delta incremental backup of only data that has changed.

Troubleshooting and Support

Appendix

Data host – the server that has been set up to host backup data

Client – the machine that BackupAssist is installed on, that sends data to the data host

SSH Authentication – For SSH communication, we use a public / private key method of authentication meaning that you will only be asked for your password once (when registering with the server), and your public key will be uploaded to the server, enabling BackupAssist to log into the server in the future in a secure, password-less manner. For more information on public / private key authentication, visit the following Wikipedia article: [Wikipedia Public Key Cryptography](#)

Daemon Authentication – In Daemon mode, your password is stored in encrypted format by BackupAssist and provided every time the backup runs. When running in Daemon mode, traffic will be unencrypted. For this reason, we recommend that you only use this closed network environments, such as LANs or WANs connected by a secure VPN. Note that VPNs inherently encrypt communication between nodes, so using Rsync in Daemon mode over a VPN is still secure.

Troubleshooting

Test connection failed: Ensure that you are able to ping your Rsync server from your BackupAssist server and you have opened up the appropriate ports on your firewall. Make sure that the username can access the path you have specified.

SSH Connection Refused: Ensure that the services 'Openssh SSHD' and 'RsyncServer' are started on the data host machine (Administrative Tools > Services). Make sure your firewall is not blocking the attempt.

Register with server failed: Ensure that you have the correct username and password as setup on your Rsync server.