BackupAssist™ v9

File Protection using rsync

Setup guide

Contents

1. Introduction

BackupAssist File Protection includes a powerful tool called rsync that can back up data across the internet to any rsync host. This guide outlines how to use rsync to protect your data.

Adding rsync backups to your backup strategy is an excellent way of insuring yourself against data loss. Critical files can be copied to a secure, offsite location, away from your office, and backing up across the internet overcomes the need to swap tapes or hard drives. Once you've selected the host where your data will be stored, no further equipment or maintenance is required. Additional storage space can be easily added to the data host as your data requirements grow, so you don't have to worry about purchasing replacement hardware. Best of all, your critical files are available whenever you need them and can be accessed from wherever you are, using BackupAssist.

Documentation

This guide explains how to set up and support an rsync backup destination. Once you have your rsync destination in place, the rsync user guide will explain how to create backups and perform restores using BackupAssist File Protection.

File Protection – Rsync user guide.

BackupAssist – Backup tab user guide

Licensing

File Protection is a standard feature included with the BackupAssist license. To back up data across the internet with rsync, requires the *Offsite Backups Add-on* license, once the initial trial period has expired. Please contact your local BackupAssist reseller or distributor for pricing information, or visit www.BackupAssist.com.

For instructions on how to activate / deactivate license keys, visit out Licensing BackupAssist page.

Overview

Rsync is an open source application used to synchronize files and directories from one location to another. BackupAssist's implementation of this technology is in the form of an rsync destination option for File Protection backups, which allows you to back up data across the internet. The data transfer is minimized because only the data that has changed is transmitted and all data packets are compressed. You can also use built-in rsync encryption to protect the data on the rsync host.

The rsync destination that you use can be either an rsync server that you maintain yourself or a third party destination that supports rsync. Third party destinations include data centers, ISPs and cloud providers. These solutions have the advantage of high availability networks with saleable storage.

BackupAssist includes a dedicated configuration screen for backups to Amazon S3 via the S3Rsync (www.S3Rsync.com) service. To backup to Amazon S3 with rsync, you will need both an Amazon account and an S3Rsync account.

2. Rsync technology

Rsync uses a checksum method to perform the bit level data transfer. Rsync checks whether any data has changed by looking at the size of a file and its modification date. If no data has changed, rsync will not transfer the data, saving time and bandwidth. If files do not match, rsync uses a checksum method called a *rolling checksum* on the changed file to see what has changed. It will then transfer only the altered or appended data within the file.

Rsync can manage data that has been inserted, added, removed and shifted, with a minimum transfer overhead. In real terms, that means efficient use of your bandwidth and data allowances. As rsync will only transfer data that has changed (and knows when file alterations have occurred), your internet backups will take less time when compared other remote backup methods such as FTP.

Terminology

In order to avoid confusion about the use of the words client, server, Windows Server and rsync server, we will use the following terms to avoid ambiguity:

Data Host - The remote machine that will be used as your backup destination.

Rsync Server - The same as the data host, but specifically referring to the machine running rsync that accepts incoming connections and data from rsync clients.

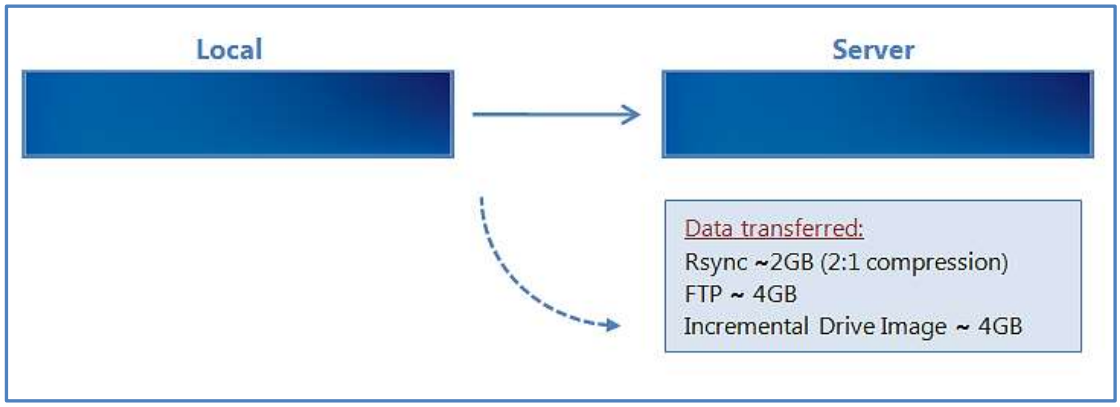
Rsync Client - A machine that contains your working data (typically a file server) that has BackupAssist installed. BackupAssist comes packaged with the rsync libraries necessary to transfer data to the rsync server during a backup.

Implementation

To help better understand how rsync transfers work, let's take a look at a hypothetical three day backup scenario.

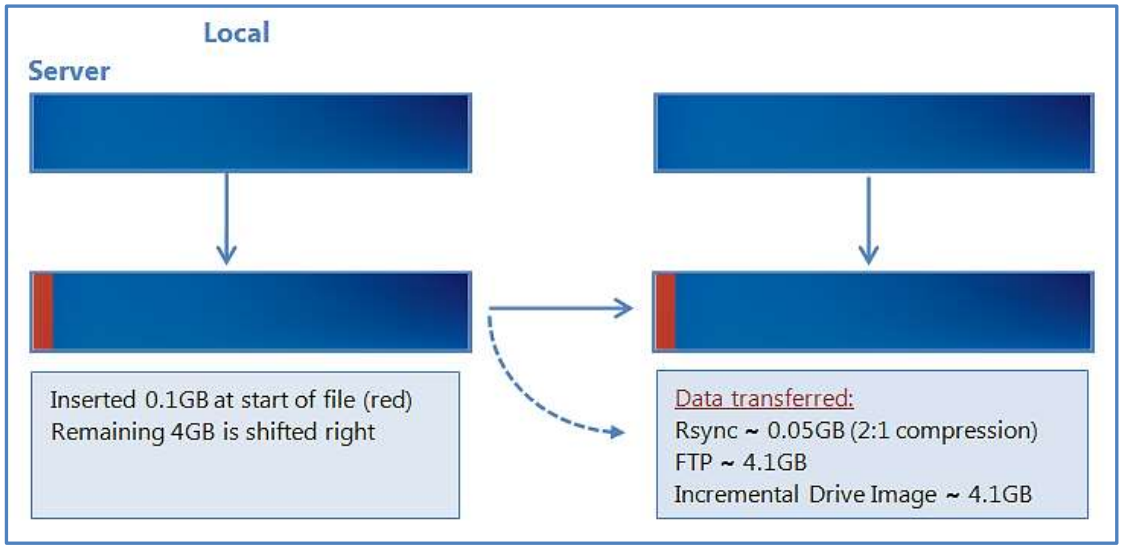
The scenario examines three different backup methods: rsync, FTP and incremental drive imaging.

Day 1: We begin with a 4GB data file backup.



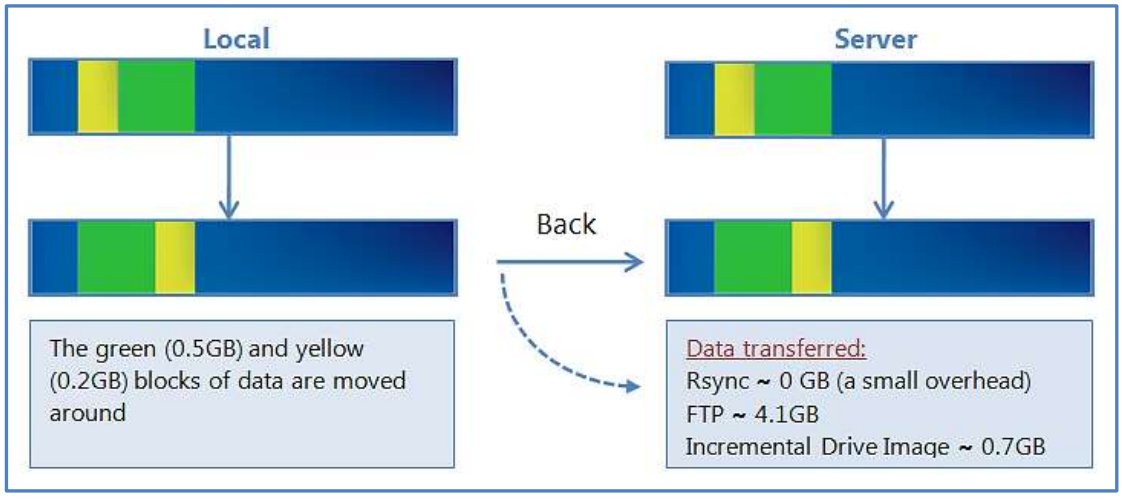
Looking at this first backup we see that for the initial data transfer there is a 100% transfer for both Incremental drive imaging and for FTP. Thanks to rsync's packet compression we see a 50% reduction in the initial transfer. Depending on your rsync server's setup this initial overhead can be removed 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 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 recognize that the data is already on the backup server and will reorganize the file with a minimal instruction file. Incremental drive imaging is also aware that the data was moved, however it must re-backup 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 backup destination selection

3. Rsync data hosts

As rsync is an open protocol, you have the option of either storing your data on a third party rsync host server, or supporting an rsync host server yourself.

For more information on how to get the most out of rsync, visit our Video Presentations page.

Third Party & S3Rsync hosting

Third party data centers, ISPs and cloud providers can support rsync backup destinations. These solutions have the advantage of high availability networks and scalable storage.

BackupAssist includes a dedicated configuration screen for backups to Amazon S3 via the s3rsync.com service. To backup to Amazon S3 with rsync you will need:

- An **Amazon AWS S3 account** (aws.amazon.com/s3/)

  In your Amazon Web Services account, you will need to obtain your Access Key ID and generate a Secret Access Key. Then you will need to create an S3 bucket to use for your backups.

  See this article for a guide to the Amazon S3 Simple Storage Service.

- An **S3Rsync account** (www.s3rsync.com)

  When you sign up for an S3Rsync (www.S3Rsync.com) account, you will be given a username and a private SSH key file. Save the SSH key file somewhere on the machine on which you wish to run BackupAssist.

- **BackupAssist**.

  You can set up your rsync backup job in BackupAssist using the S3Rsync *Destination* selection.

Do-it-yourself hosting: overview

Any rsync Server such as an rsync-enabled NAS device, a Windows Server or a Unix machine can be used to store backups using rsync. The do-it-yourself approach has the advantage of keeping data in your control and a lack of monthly hosting fees.

Rsync servers can be one of two types:

- **Rsync over SSH** (preferred) runs rsync via a secure shell (SSH, port 22) which means all traffic over the internet is encrypted. User access control is modified by editing user accounts on the server.

- **Daemon mode** runs rsync as a normal TCP/IP service. User access control is modified by editing the rsync.conf file. Internet traffic is not encrypted.

  To learn more, review our online article Configuring BackupAssist for rsync without SSH, under the section, *Altering the rsyncd.conf file*.

In the following sections, 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.

Do-it-yourself hosting: Windows rsync host

To set up a Windows machine to act as an rsync server, you will first need to install both SSH and rsync on your Windows Server. We recommend CopSSH and cwRsyncServer. An installer for each can be found on our website by visiting http://www.backupassist.com/rsync.

Prerequisites:

- Windows Server 2008, 2012 or 2016 machine with network connectivity and space to store backup data.
- Windows Server 2008, 2012 or 2016 are highly recommended because of their support for both backup histories and single-instance store in rsync backup solutions.
- Windows Small Business Servers (SBS) should not be used as rsync hosts.
- 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 a location of your choice.
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 finished cwRsync will be present on your system.

Installing CopSSH:

1. Run the CopSSH installer.
2. Continue through the installation wizard, installing the package to a location of your choice.
3. During the installation you will be presented with the popup on the right. We suggest leaving the SvcCOPSSH account as is. Write down the password provided.
4. Click *Install* to complete the process of installing CopSSH on your system.
5. During the Activate user part of the installation, you will be presented with a popup showing the service status and any active connections. At any time after the install 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 can register an rsync client.
6. 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.

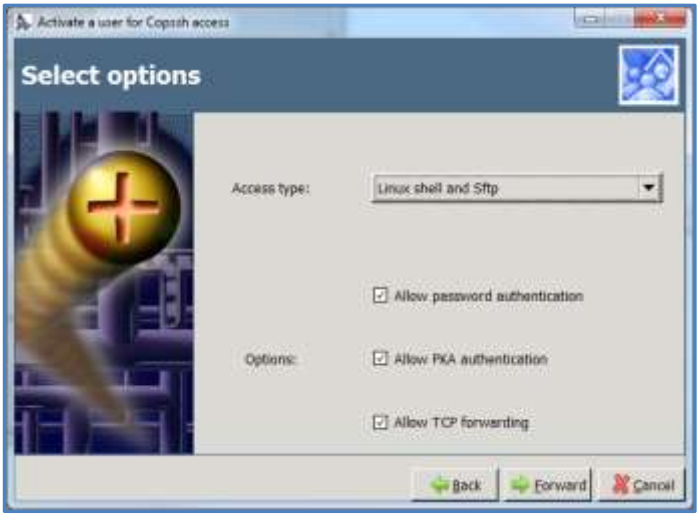
1. In the *Start* menu, under *All Programs* -> CopSSH, select. The CopSSH Control Panel will open.
2. To start the process to activate a user, click on the *Users* tab across the top of the user interface.
3. Click on the *Add* button to bring up the wizard to activate a user.

DO NOT ACTIVATE USING 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.

4. Click *Forward* on the opening screen.

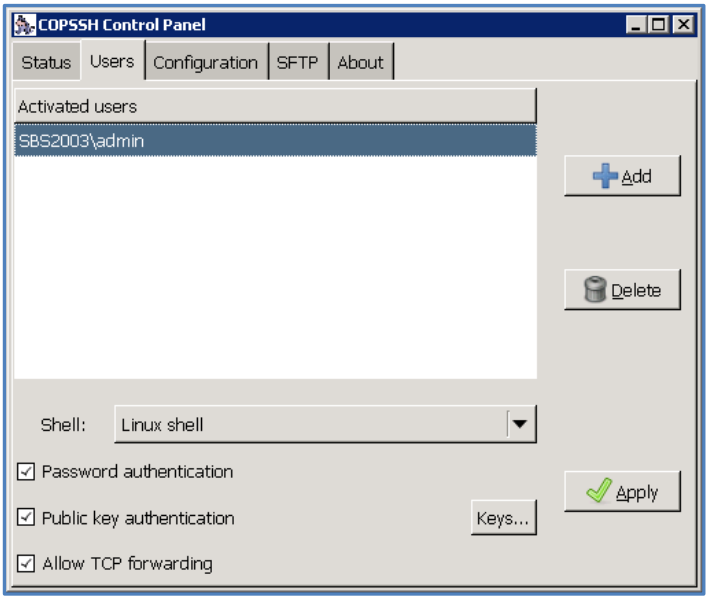
5. On the second screen, select the Domain and type in the user which you wish to activate.
   Click *Forward* once complete (admin is a manually created account we'll use for this example).

6. Change the *Access Type* to *Linux Shell and Sftp* using the drop-down menu.
   Leave all *Options* enabled as they are by default.



7. 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\user*.

The location of this directory can be changed by editing the file *C:\Program Files\ICW\etc\passwd*.

> **Note**: If you uninstall the rsync server, be aware that the Windows service users *SvcCOPSSH* and *SvcCWRSYNC* are not removed. So if you then re-install the CWRsync Server package the Windows users cannot be recreated because the passwords will 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.

Do-it-yourself hosting: Linux rsync 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.

Note: You can choose to run rsync as a daemon on your Linux server. (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 below.

To determine if your system has the prerequisites installed, log into your system, start a shell and 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.

You should use your distribution's software package manager to install these packages, if they are not already installed. Most commonly they can be found under the *Server* or *Security* categories. 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, otherwise 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

Do-it-yourself hosting: NAS rsync host

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 that directory.
- If your NAS requires a password to connect to the rsync service, you will need 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).

The options vary from device to device. You will need to consult your manual to setup the destination.

Below is a list of NAS vendors that support rsync.

QNAP : drobo : NETGEAR : Synology > Click on any of the vendor below to go to their website.

Do-it-yourself hosting: host configuration tips

The following table contains tips designed to address common mistakes and highlight considerations that will help you set up an rsync host.

Tip	Description
Make sure it's rsync compatible	When you select hardware to use as an rsync server, make sure the hardware can support the rsync protocol. If you select a Windows system, it must be able to run cwRsync. A NAS device must have rsync specified as one of the protocols supported. If in doubt, ask your hardware vendor for confirmation.
Processing speed is important!	Rsync can be a very processing intensive protocol - it uses checksums that calculate what data needs to be transferred. A lot of NAS devices come with lower range CPUs built-in. This will affect the overall time taken to complete an rsync backup.
Ensure there is plenty of disk space available	Although you may think you have enough disk space available when you first implement your rsync solution, a common cause of rsync problems is that the storage space eventually runs out. Some of the BackupAssist backup schemes are designed to retain significant amounts of data – meaning the space you have can be used up faster than you expect! Running out of disk space is a common problem and it can cause a lot of problems when it occurs. For this reason, the available storage space on your rsync host should be monitored.
Make sure you set the correct backup path	Some NAS devices contain a boot partition (similar to Windows Server 2008R2). Sometimes, if you enter the incorrect path your rsync backup will write to this boot partition – which could in turn cause major issues with your backup and hardware.
Seed your backup	If you're planning on using a NAS device, you can run your seed backup by connecting your NAS device directly to the local network. This avoids having to seed to a USB drive, and then running the seed to the NAS device in a two-step process (saving you a lot of time).
Double check permissions	Even though you are logged in as a Domain Admin, most NAS devices require users to be set up locally within the unit and have permissions configured locally as well. If you receive permission issues, this is usually the reason as to why.

4. Rsync backup considerations

The performance and flexibility of backing up across the internet can depend on how rsync is implemented. Below are some key considerations when planning your rsync backup solution.

Setup considerations

Backup user accounts

Rsync backup jobs require a BackupAssist administrator account with read access to the data source. This is set up using the *Backup user identity*, option in the *Settings* tab. The backup job will also need an rsync host account with read-write access to the rsync destination. This is enabled on the host server, and entered in the rsync destination screen.

Data Seeding

Rsync backups are incremental backups. The first time you perform your backup, no data will exist on the data host so a full backup will be required. Seeding your backup via an internet connection may not be practical, so two methods are provided to seed your data host.

Backup source & frequency

Run your rsync job regularly. Regular daily backups will ensure that you keep your data transfer to a minimum and your data up-to-date.

Simultaneous backups

If you have a large number of backup jobs sending data to a host at the same time, the connections may become unreliable. It is recommended that you limit the host connections 5 at a time. Depending on storage requirements and the bandwidth available, you may increase this number with caution.

Exchange databases and SQL databases

VSS applications including Exchange, SQL and Hyper-V, can be backed up to an rsync destination using File Protection. Simply choose the VSS application that you want to back up from the list of detected applications. You can even drill down and choose individual components (databases, storage groups, etc.) to backup. For Hyper-V, we recommended System Protection backups, which do not support the rsync destination but do support granular Hyper-V guest restores.

Synchronizing drive images using rsync

Rsync is a destination for File Protection backups. It is possible for the data source to be a System Protection image backup, but this solution is not recommended because significant performance issues that can arise. If you want to back up important files to an rsync host, the best way is to back up those files using File Protection directly. Continue to create your image backups, but back up the important files independently using a File Protection rsync backup job.

We also advise against using File Protection's rsync, to transfer File Archiving backups to an rsync host. This is because rsync uses a checksum method to perform the bit level data transfer. Rsync checks whether any data has changed by looking at the file size and modification date. This is fast and simple on a regular file system, but if you have a very large single archive file (>100 GB) it will take much longer to complete, even if only a small element has changed.

Technology considerations

Using a NAS device as a 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 reduced performance if you are backing up large files (several GB or larger in a single file). The example below illustrates the difference in backup time for a dedicated QNAP NAS, 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	Ubuntu 9.04 desktop with rsync
Initial backup	7 hours 55 minutes	1 hours 22 minutes
Second backup	4 hours 57 minutes	0 hours 35 minutes

Data compression and encryption

BackupAssist supports encryption and compression on the server. 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.

Rsync for BackupAssist uses four types of compression:

- Effective transfer compression by only sending changed data.
- All data packets are compressed and encrypted during transfer.
- Single Instance Store (SIS) uses hard link technology to prevent the same files from being stored more than once across backups on your host.
- The source data is encrypted and compressed in an 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).

Single-Instance store

File Protection backups cannot use single-instance store when the backup is saved on a ReFS formatted rsync destination. This means all the data will be backed up each time the backup job runs.

Preservation of file attributes

Because rsync works on top of the Cygwin Unix emulation layer, it does not recognize Windows file attributes (e.g. read-only, hidden), NTFS security attributes (i.e. access control lists), NTFS alternate data streams or file creation times. The only file system attribute preserved when using rsync to transfer data is the Last modified time attribute.

BackupAssist's implementation of rsync overcomes this limitation by having the option to store NTFS metadata on the backup destination. This option is enabled in the *Manage* screen under *Rsync options*. This is checked by default for new jobs created in BackupAssist. If enabled, NTFS 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 Integrated Restore Console. So while these attributes are not "preserved" on the files backed up to your rsync destination, they will still be restored.

This table outlines what attributes are preserved with the NTFS metadata option:

File attributes at destination	Preserved
Windows File Attributes	✘
Creation time	✘
Last access time	✘
Last modified time	✓
NTFS security (Access Control Lists)	✓
NTFS alternate data streams (ADSs)	✓

Using rsync for files and directories

Rsync performs best when working directly on the file system, backing up normal files and directories.

Below is an example of how rsync performs:

- 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 what changed. 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.

Compare this to a scenario where you use a File Protection backup job to transfer an image file created by a System Protection backup, to an rsync destination. In this example, the VHD is 50GB.

Rsync will detect that the single VHD 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 VHD 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 the operate rsync on the underlying file system rather than a backup of that file system.

File size and the number of files

In theory, there is no limit to the number of files or directories that you can rsync. Even though rsync only transfers the data that has changed, it still must read all of the data in the file set to check what data has changed. This makes rsync internet backups a disk/CPU intensive operation that can take 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 BackupAssist's File Archiving) 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.

5. Support and Resources

These resources can be used to help troubleshoot common rsync backup problems.

Troubleshooting FAQ

Test connection failed

Ensure that you are able to ping your rsync server from your BackupAssist server and that 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 set up on your rsync server.

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 an 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 closed network environments, such as LANs or WANs connected by a secure VPN. VPNs inherently encrypt communication between nodes, so using rsync in Daemon mode over a VPN is still secure.