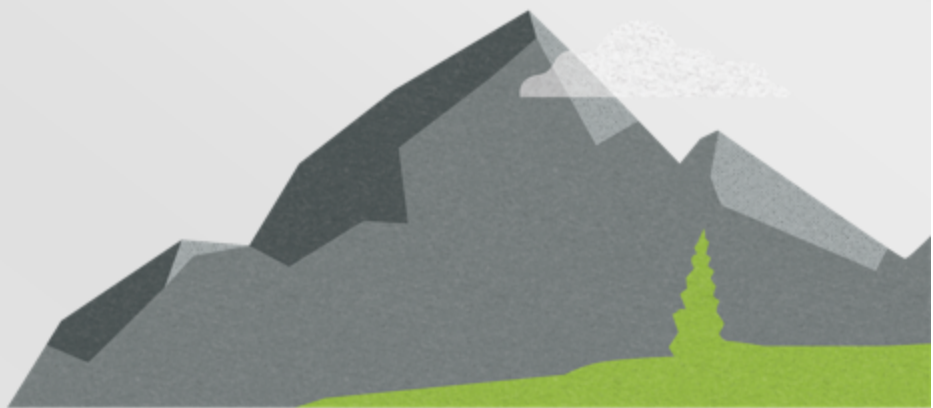


Carbonite Availability for Linux

User's Guide



Notices

Carbonite Availability for Linux User's Guide Version 8.1.1, Thursday, April 5, 2018

If you need technical assistance, you can contact CustomerCare. All basic configurations outlined in the online documentation will be supported through CustomerCare. Assistance and support for advanced configurations may be referred to a Pre-Sales Systems Engineer or to Professional Services.

Man pages are installed and available on Carbonite Availability and Carbonite Move Linux servers. These documents are bound by the same Carbonite license agreement as the software installation.

This documentation is subject to the following: (1) Change without notice; (2) Furnished pursuant to a license agreement; (3) Proprietary to the respective owner; (4) Not to be copied or reproduced unless authorized pursuant to the license agreement; (5) Provided without any expressed or implied warranties, (6) Does not entitle Licensee, End User or any other party to the source code or source code documentation of anything within the documentation or otherwise provided that is proprietary to Carbonite, Inc.; and (7) All Open Source and Third-Party Components ("OSTPC") are provided "AS IS" pursuant to that OSTPC's license agreement and disclaimers of warranties and liability.

Carbonite, Inc. and/or its affiliates and subsidiaries in the United States and/or other countries own/hold rights to certain trademarks, registered trademarks, and logos. Hyper-V and Windows are registered trademarks of Microsoft Corporation in the United States and/or other countries. Linux is a registered trademark of Linus Torvalds. vSphere is a registered trademark of VMware. All other trademarks are the property of their respective companies. For a complete list of trademarks registered to other companies, please visit that company's website.

© 2018 Carbonite, Inc. All rights reserved.

Contents

Chapter 1 Carbonite Availability overview	6
Core operations	7
Supported configurations	10
Replication capabilities	17
Chapter 2 Carbonite Availability clients	19
Replication Console for Linux for files and folders jobs	20
Logging on and off	21
Using Replication Console for Linux workspaces	23
Clearing stored security credentials	24
Server settings	25
Identifying a server	26
Licensing a server	28
Configuring server startup options	31
Configuring network communication properties for a server	33
Queuing data	35
Configuring source data processing options	38
Configuring target data processing options	40
Specifying the Carbonite Availability database storage files	41
Specifying file names for logging and statistics	42
E-mailing system messages	44
Failover for Linux console for files and folders jobs	47
Setting the frequency of Failover for Linux console refreshes	48
Carbonite Replication Console for full server and full server to ESX jobs	49
Carbonite Replication Console requirements	51
Console options	52
Managing servers	55
Adding servers	64
Providing server credentials	66
Viewing server details	67
Editing server properties	69
General server properties	70
Server licensing	71
E-mail notification configuration	73
Viewing server logs	75
Managing VMware servers	77
Managing snapshots	78
Chapter 3 Files and folders protection	79
Files and folders requirements	80
Creating a files and folders job	83
Establishing a data connection using the automated Connection Wizard	84
Creating a replication set	86
Establishing a connection manually using the Connection Manager	89
Establishing a connection across a NAT or firewall	93
Simulating a connection	95
Protection monitoring	96
Monitoring a data workload	97

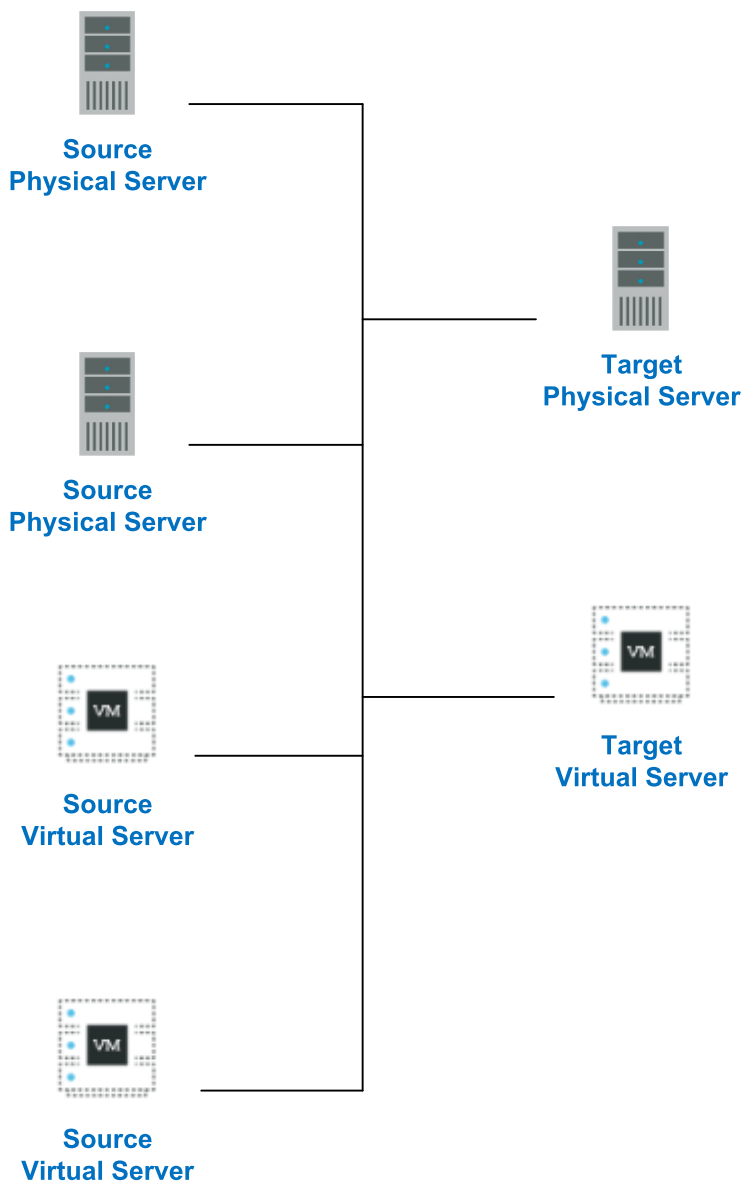
Viewing the Carbonite Availability log file through the Replication Console for Linux	104
Configuring the properties of the Carbonite Availability log file	106
Configuring the properties of the statistics file	107
E-mailing system messages	108
Connections	111
Pausing and resuming target processing	112
Disconnecting a connection	113
Mirroring	114
Stopping, starting, pausing, or resuming mirroring	115
Mirroring automatically	117
Removing orphan files	119
Replication	121
Replication sets	122
Creating a replication set	124
Creating or modifying replication rules manually	127
Selecting a block device for replication	129
Modifying a replication set	130
Renaming and copying a replication set	131
Calculating replication set size	132
Exporting and importing a replication set	134
Deleting a replication set	135
Starting replication	136
Inserting tasks during replication	137
Verification	138
Verifying manually	139
Verifying on a schedule	140
Configuring the verification log	142
Data transmission	144
Stopping, starting, pausing, and resuming transmission	145
Scheduling data transmission	145
Limiting transmission bandwidth	150
Compressing data for transmission	152
Failover	154
Configuring failover monitoring	155
WAN considerations	158
Protecting NFS exports	160
Protecting Samba shares	161
Editing failover monitoring configuration	162
Monitoring failover monitoring	163
Failing over	166
Removing failover monitoring configuration	166
Failback and restoration	167
Restoring then failing back	168
Failing back then restoring	173
Chapter 4 Full server protection	176
Full server requirements	177
Creating a full server job	184
Managing and controlling full server jobs	202

Viewing full server job details	210
Validating a full server job	214
Editing a full server job	215
Viewing a full server job log	217
Failing over full server jobs	219
Reversing full server jobs	221
Chapter 5 Full server to ESX protection	222
Full server to ESX requirements	223
Creating a full server to ESX job	229
Managing and controlling full server to ESX jobs	255
Viewing full server to ESX job details	263
Validating a full server to ESX job	267
Editing a full server to ESX job	268
Viewing a full server to ESX job log	270
Failing over full server to ESX jobs	272
Reversing protection after failover for full server to ESX jobs	274
Chapter 6 DTSetup	275
Running DTSetup	276
Setup tasks	277
Activating your server	278
Modifying security groups	279
Configuring server settings	280
Configuring driver performance settings	281
Starting and stopping the service	282
Starting DTCL	283
Viewing documentation and troubleshooting tools	284
DTSetup menus	285
Chapter 7 Security	286
Adding users to the security groups	287
Chapter 8 Special network configurations	288
Firewalls	289
IP and port forwarding	290

Chapter 1 Carbonite Availability overview

Carbonite Availability ensures the availability of critical workloads. Using real-time replication and failover, you can protect data or entire servers, running on physical or virtual servers.

You identify what you want to protect on your production server, known as the source, and replicate that to a backup server, known as the target. The target server, on a local network or at a remote site, stores a replica copy of the data from the source. Carbonite Availability monitors any changes to the source and sends the changes to the replica copy stored on the target server. By replicating only the file changes rather than copying an entire file, Carbonite Availability allows you to more efficiently use resources.



Core operations

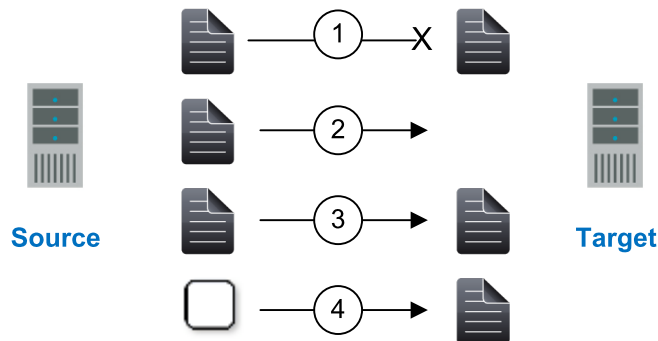
Carbonite Availability performs three basic types of operations.

- *Mirroring* on page 7—The initial copy or subsequent resynchronization of selected data
- *Replication* on page 8—The on-going capture of byte-level file changes
- *Failover* on page 9—The ability to stand-in for a server, in the event of a failure

Mirroring

Mirroring is the process of transmitting user-specified data from the source to the target so that an identical copy of data exists on the target. When Carbonite Availability initially performs mirroring, it copies all of the selected data, including file attributes and permissions. Mirroring creates a foundation upon which Carbonite Availability can efficiently update the target server by replicating only file changes.

If subsequent mirroring operations are necessary, Carbonite Availability can mirror specific files or blocks of changed data within files. By mirroring only files that have changed, network administrators can expedite the mirroring of data on the source and target servers. Mirroring has a defined end point when all of the selected files from the source have been transmitted to the target. When a mirror is complete, the target contains a copy of the source files at that point in time.

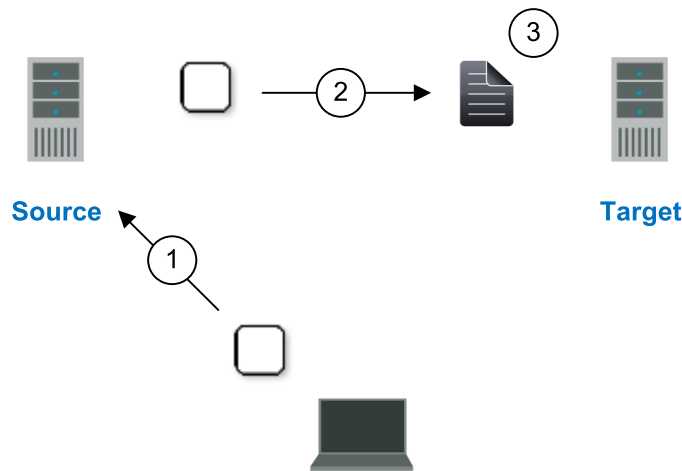


1. Identical files are not mirrored.
2. New files are mirrored.
3. Different files can be mirrored.
4. Checksums can calculate blocks of data to be mirrored.

Replication

Replication is the real-time transmission of file changes. Unlike other related technologies, which are based on a disk driver or a specific application, the Carbonite Availability replication process operates at the file system level and is able to track file changes independently from the file's related application. In terms of network resources and time, replicating changes is a more efficient method of maintaining a real-time copy of data than copying an entire file that has changed.

After a source and target have been connected through Carbonite Availability, file system changes from the user-defined data set are tracked. Carbonite Availability immediately transmits these file changes to the target server. This real-time replication keeps the data on the target up-to-date with the source and provides high availability and disaster recovery with minimal data loss. Unlike mirroring which is complete when all of the files have been transmitted to the target, replication continuously captures the changes as they are written to the source. Replication keeps the target up-to-date and synchronized with the source.



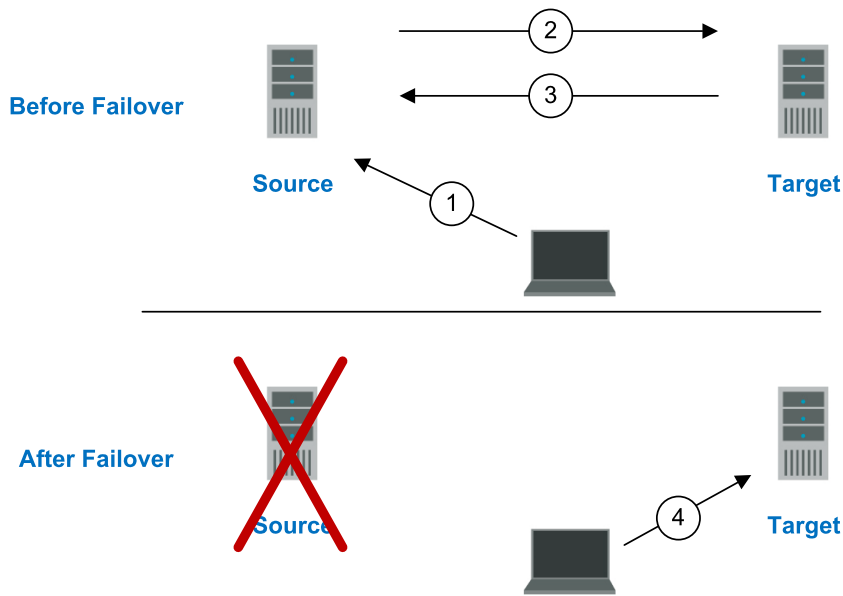
1. A user or application updates part of a file.
2. Only the changed portion of the file is replicated to the target.
3. An up-to-date copy of the file is maintained on the target.

Failover

Failover is the process in which a target stands in for a failed source. As a result, user and application requests that are directed to the failed source are routed to the target.

Carbonite Availability monitors the source status by tracking requests and responses exchanged between the source and target. When a monitored source does not respond to the target's requests, Carbonite Availability assumes that the server has failed. Carbonite Availability then prompts the network administrator to initiate failover, or, if configured, it occurs automatically. The failover target assumes the identity of the failed source, and user and application requests destined for the source server or its IP address(es) are routed to the target.

When partnered with the Carbonite Availability data replication capabilities, failover routes user and application requests with minimal disruption and little or no data loss.



1. User and application requests are sent to the source name or IP address.
2. Data on the source is mirrored and replicated to the target.
3. The target monitors the source for failure.
4. In the event the source fails, the target stands in for the source. User and application requests are still sent to the source name or IP address, which are now running on the target.

Supported configurations

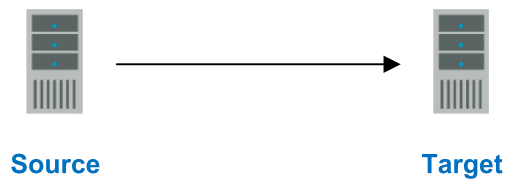
Carbonite Availability is an exceptionally flexible product that can be used in a wide variety of network configurations. To implement Carbonite Availability effectively, it is important to understand the possible configuration options and their relative benefits. Carbonite Availability configurations can be used independently or in varying combinations.



Not all types of jobs support all of these configurations. See the requirements of each job type to determine which configurations are supported.

- *One to one, active/standby* on page 11
- *One to one, active/active* on page 12
- *Many to one* on page 13
- *One to many* on page 14
- *Chained* on page 15
- *Single server* on page 16

One to one, active/standby



Description

One target server, having no production activity, is dedicated to support one source server. The source is the only server actively replicating data.

Applications

- This configuration is appropriate for offsite disaster recovery, failover, and critical data backup. This is especially appropriate for critical application servers.
- This is the easiest configuration to implement, support, and maintain.

Considerations

- This configuration requires the highest hardware cost because a target server is required for every source server.
 - You must pause the target when backing up database files on the target.
-

One to one, active/active



Description

Each server acts as both a source and target actively replicating data to each other

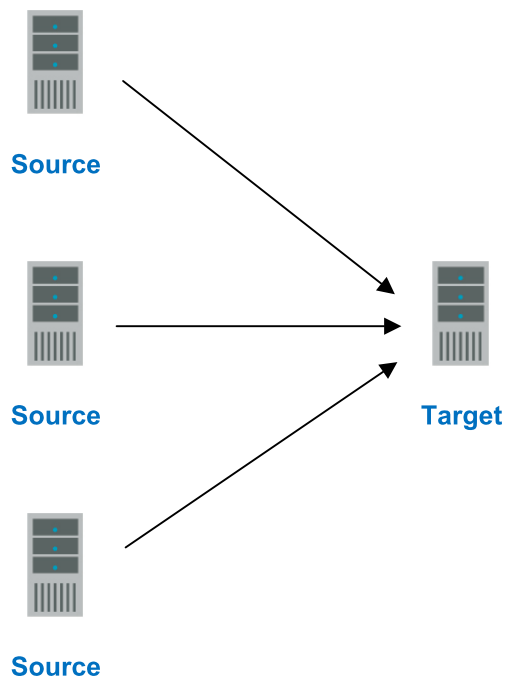
Applications

This configuration is appropriate for failover and critical data backup. This configuration is more cost-effective than the Active/Standby configuration because there is no need to buy a dedicated target server for each source. In this case, both servers can do full-time production work.

Considerations

- Coordination of the configuration of Carbonite Availability and other applications can be more complex than the one to one active/standby configuration.
- During replication, each server must continue to process its normal workload.
- Administrators must avoid selecting a target destination path that is included in the source's protected data set. Any overlap will cause an infinite loop.
- To support the production activities of both servers during failover without reducing performance, each server should have sufficient disk space and processing resources.
- Failover and failback scripts must be implemented to avoid conflict with the existing production applications.
- You must pause the target when backing up database files on the target.

Many to one



Description

Many source servers are protected by one target server.

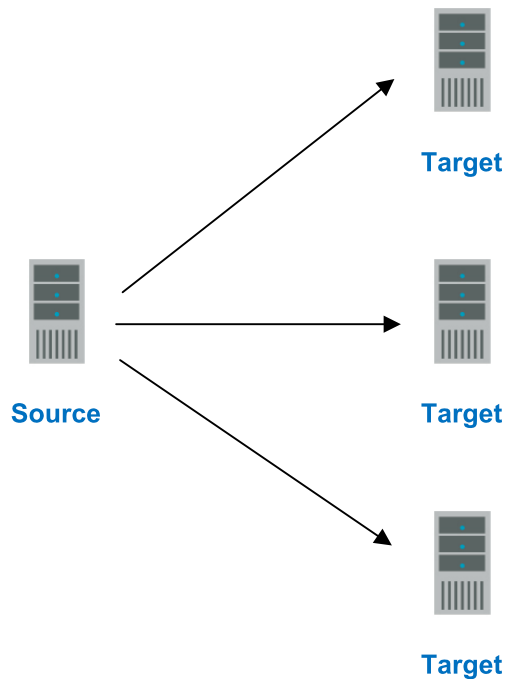
Applications

This configuration is appropriate for offsite disaster recovery. This is also an excellent choice for providing centralized tape backup because it spreads the cost of one target server among many source servers.

Considerations

- The target server must be carefully managed. It must have enough disk space and RAM to support replication from all of the source systems. The target must be able to accommodate traffic from all of the servers simultaneously.
- If using failover, scripts must be coordinated to ensure that, in the event that the target server stands in for a failed server, applications will not conflict.
- You must pause the target when backing up database files on the target.

One to many



Description

One source server sends data to multiple target servers. The target servers may or may not be accessible by one another.

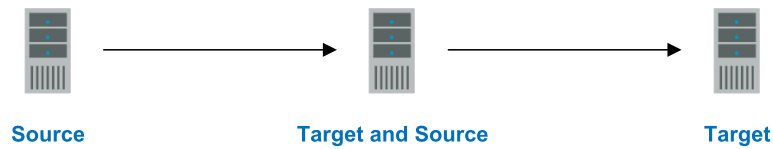
Applications

This configuration provides offsite disaster recovery, redundant backups, and data distribution. For example, this configuration can replicate all data to a local target server and separately replicate a subset of the mission-critical data to an offsite disaster recovery server.

Considerations

- Updates are transmitted multiple times across the network. If one of the target servers is on a WAN, the source server is burdened with WAN communications.
- You must pause the target when backing up database files on the target.
- If you failover to one of the targets, the other targets stop receiving updates.

Chained



Description

The source servers send replicated data to a target server, which acts as a source server and sends data to a final target server, which is often offsite.

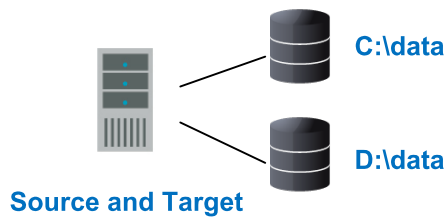
Applications

This is a convenient approach for integrating local high availability with offsite disaster recovery. This configuration moves the processing burden of WAN communications from the source server to the target/source server. After failover in a one to one, many to one, or one to many configuration, the data on the target is no longer protected. This configuration allows failover from the first source to the middle machine, with the third machine still protecting the data.

Considerations

- The target/source server could become a single point of failure for offsite data protection.
- You must pause the target when backing up database files on the target.

Single server



Description

Source and target components are loaded on the same server allowing data to be replicated from one location to another on the same volume or to a separate volume on the same server. These could be locally attached SCSI drives or Fibre Channel based SAN devices.

Applications

This configuration is useful upgrading storage hardware while leaving an application online. Once the data is mirrored, you can swap the drive in the disk manager. If the source and target copies of the data are located on different drives, this configuration supports high availability of the data in the event that the source hard drive fails.

Considerations

- This configuration does not provide high availability for the entire server.
- This configuration must be configured carefully so that an infinite loop is not created.
- This configuration should be limited to a single Carbonite Availability job.
- This configuration should be used sparingly. If possible, you should attach the target volumes to another server and use a one to one configuration.

Replication capabilities

Carbonite Availability replicates all file and directory data in the supported Linux file systems. Carbonite Availability does not replicate items that are not stored on the file system, such as pseudo-file systems like /proc and /sys. In addition, note the following.

- Carbonite Availability is compatible with NFS and Samba services as long as they are mounted on top of Carbonite Availability. (The mount must be at the origination point, not a remote mounted point.) Additionally, NFS and Samba should be started after the Double-Take service.
- If you select data stored on a recursive mount point for replication, a mirror will never finish. Carbonite Availability does not check for data stored on recursive mount points.
- If any directory or file contained in your replication set specifically denies permission to the account running the Double-Take service, the attributes of the file on the target will not be updated because of the lack of access.
- Sparse files will become full size, zero filled files on the target.
- If you are using soft links, keep in mind the following.
 - If a soft link to a directory is part of a replication set rule's path above the entry point to the replication set data, that link will be created on the target as a regular directory if it must be created as part of the target path.
 - If a soft link exists in a replication set (or is moved into a replication set) and points to a file or directory inside the replication set, Carbonite Availability will remap the path contained in that link based on the Carbonite Availability target path when the option RemapLink is set to the default value (1). If RemapLink is set to zero (0), the path contained in the link will retain its original mapping.
 - If a soft link exists in a replication set (or is moved into a replication set) and points to a file or directory outside the replication set, the path contained in that link will retain its original mapping and is not affected by the RemapLink option.
 - If a soft link is moved out of or deleted from a replication set on the source, that link will be deleted from the target.
 - If a soft link to a file is copied into a replication set on the source and the operating system copies the file that the link pointed to rather than the link itself, then Carbonite Availability replicates the file copied by the operating system to the target. If the operating system does not follow the link, only the link is copied.
 - If a soft link to a directory is copied into a replication set on the source and the operating system copies the directory and all of its contents that the link pointed to rather than the link itself, then Carbonite Availability replicates the directory and its contents copied by the operating system to the target. If the operating system does not follow the link, only the link is copied.
 - If any operating system commands, such as chmod or chown, is directed at a soft link on the source and the operating system redirects the action to the file or directory which the link references, then if the file or directory referenced by the link is in a replication set, the operation will be replicated for that file to the target.
 - The operating system redirects all writes to soft links to the file referenced by the link. Therefore, if the file referenced by the symbolic link is in a replication set, the write operation will be replicated to the target.

- If you are using hard links, keep in mind the following.
 - If a hard link exists (or is created) only inside the replication set on the source, having no locations outside the replication set, the linked file will be mirrored to the target for all locations and those locations will be linked if all link locations on the target exist on the same partition.
 - If a hard link crosses the boundaries of a replication set on the source, having locations both inside and outside the replication set, the linked file will be mirrored to the target for only those locations inside the replication set on the source, and those locations will be linked on the target if all link locations exist on the same partition.
 - If a hard link is created on the source linking a file outside the replication set to a location inside the replication set, the linked file will be created on the target in the location defined by the link inside the replication set and will be linked to any other locations for that file which exist inside the replication set.
 - If any hard link location is moved from outside the replication set into the replication set on the source, the link will not be replicated to the target even if other link locations already exist inside the replication set, but the linked file will be created on the target in the location defined by the link.
 - If any hard link location existing inside the replication set is moved within the replication set on the source, the move will be replicated to the target and the link will be maintained if the new link location does not cross partitions in the target path.
 - If any hard link location existing inside the replication set is moved out of the replication set, that file or linked location will be deleted on the target.
 - If a hard linked file is copied from any location inside or outside the replication set to a location inside the replication set on the source, the copy will be replicated to the target.
 - If a hard linked file has a location in the replication set and any of the operating system commands, such as `chmod` or `chown`, are directed at that file from a location inside the replication set, the modification to the file will be replicated to the target. Operations on hard links outside of the replication set are not replicated.
 - If a hard linked file has a location in the replication set and a write operation is directed at that file from inside the replication set, the write operation will be replicated to the target. Operations on hard links outside of the replication set are not replicated.
 - If any hard link location existing inside the replication set is deleted on the source, that file or linked location will be deleted from the target.

Chapter 2 Carbonite Availability clients

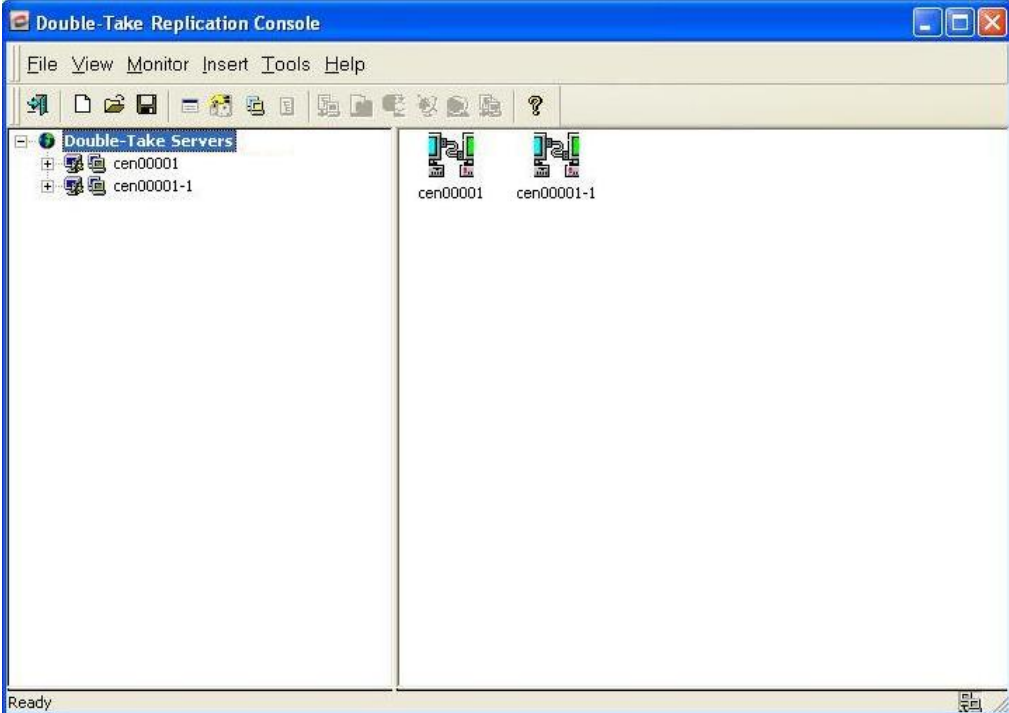
Carbonite Availability for Linux has different clients for different job types.

- **Files and folders jobs**—Files and folders jobs will use the Replication Console for Linux and the Failover for Linux console to control and manage your connections and failover. These client installations are not detailed in the *Carbonite Availability and Carbonite Move Installation, Licensing, and Activation* document. You can install these clients by selecting the **Install Carbonite Replication for Linux Management Client** link from the installation landing page. Follow the on-screen installation instructions. After the installation is complete, both clients can be started from the Windows **Start** menu. You can also launch the Failover for Linux console from the **Tools** menu in the Replication Console for Linux. Linux files and folders jobs can also use a DTCL scripting language to control and manage connections and failover. For more information, see the *DTCL Scripting Guide*.
 - *Replication Console for Linux for files and folders jobs* on page 20
 - *Failover for Linux console for files and folders jobs* on page 47
- **Full server and full server to ESX jobs**—Full server and full server to ESX jobs use the Carbonite Replication Console to control and manage the jobs and failover. This client installation is detailed in the *Carbonite Availability and Carbonite Move Installation, Licensing, and Activation* document. After the installation is complete, the console can be started from the Windows **Start** menu. Linux full server and full server to ESX jobs can also use PowerShell scripting to control and manage these jobs types. For more information, see the *PowerShell Scripting Guide*.
 - *Carbonite Replication Console for full server and full server to ESX jobs* on page 49

Replication Console for Linux for files and folders jobs

Start the Carbonite Availability Replication Console for Linux by selecting **Carbonite, Replication, Carbonite Replication Console for Linux** from your **Programs, All Programs, or Apps**, depending on your operating system.

From the Replication Console for Linux, you can manage, monitor, and control your Carbonite Availability connections. The Replication Console for Linux is a two pane view. The views in the panes change depending on what is highlighted. For example, when the root of the tree in the left pane is selected, all of the machines in your environment running Carbonite Availability are displayed in the right pane. If you expand the tree in the left pane and select a server, any connections for that server are displayed in the right pane.



Logging on and off

To ensure protection of your data, Carbonite Availability offer multi-level security using native operating system security features. Privileges are granted through membership in user groups defined on each machine running Carbonite Availability. To gain access to a particular Carbonite Availability source or target, the user must provide a valid operating system user name and password and the specified user name must be a member of one of the Carbonite Availability security groups. Once a valid user name and password has been provided and the Carbonite Availability source or target has verified membership in one of the Carbonite Availability security groups, the user is granted appropriate access to the source or target and the corresponding features are enabled in the client. Access to Carbonite Availability is granted on one of the following three levels.

- **Administrator Access**—All features are available for that machine.
- **Monitor Access**—Servers and statistics can be viewed, but functionality is not available.
- **No Access**—Servers appear in the clients, but no access to view the server details is available.

Use the following instructions when logging on and off of a server.

1. Highlight a machine on the left pane of the Replication Console for Linux. By double-clicking the machine name, Carbonite Availability automatically attempts to log you on to the selected machine using the ID that you are currently logged on with. Verify your access by the resulting icon.
2. If you have no access, the Logon dialog box will automatically appear. If you have monitor access or want to log on with a different username, right-click the machine name and select **Logon**.



3. Specify your **Username, Password, Domain**, and whether you want your password saved.
4. Click **OK** and verify your access by the resulting icon and log on again if necessary.



When logging in, the user name, password, and domain are limited to 100 characters.

If your license key is missing or invalid, you will be prompted to open the Server Properties **General** tab to add or correct the key. Select **Yes** to open the Server Properties dialog box or select **No** to continue without adding a license key.

If the login does not complete within 30 seconds, it is automatically canceled. If this timeout is not long enough for your environment, you can increase it by adjusting the **Communication Timeout** on the **Configuration** tab of the Replication Console for

Linux properties. Select **File, Options**, from the Replication Console for Linux to access this screen.

Carbonite Availability uses ICMP pings to verify server availability during the login process. If your Carbonite Availability server is across a router or firewall that has ICMP pings disabled, you will need to disable the Carbonite Availability ICMP ping verification. To do this, select **File, Options**, from the Replication Console for Linux and disable **Use ICMP to verify server availability**.

Administrator rights

This icon is a computer with a gear and it indicates the Carbonite Availability security is set to administrator access.

Monitor rights

This icon is a computer with a magnifying glass and it indicates the Carbonite Availability security is set to monitor only access.

No rights

This icon is a lock and it indicates the Carbonite Availability security is set to no access.

5. To log off of a Carbonite Availability machine, right-click the machine name on the left pane of the Replication Console for Linux and select **Logout**.

Using Replication Console for Linux workspaces

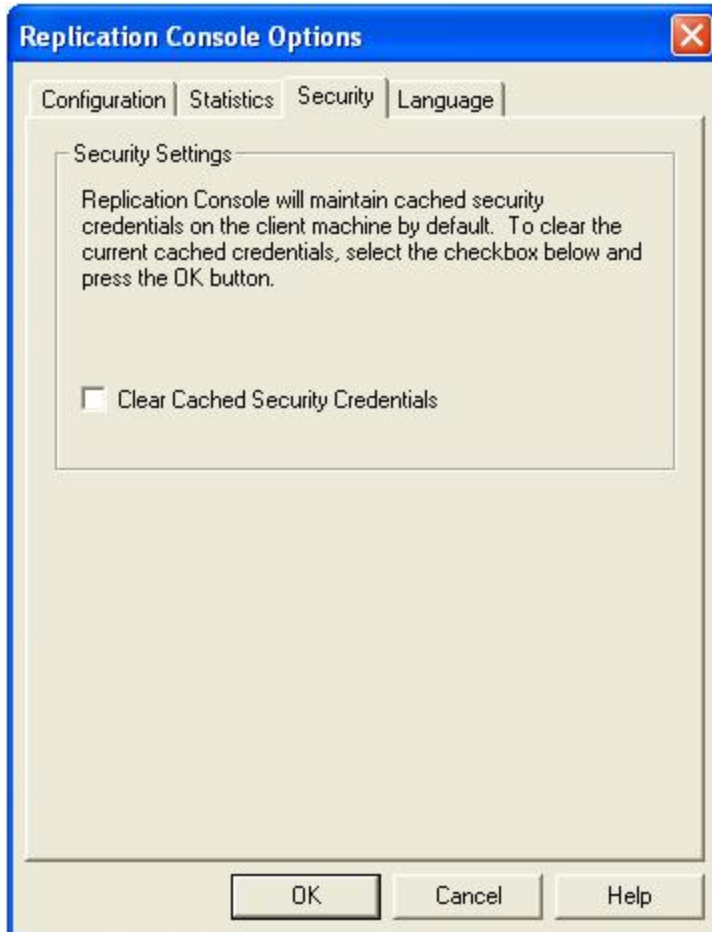
The Replication Console for Linux workspace contains the display of the panes of the Replication Console for Linux and any servers that may have been inserted. Multiple workspaces can be used to help organize your environment or to view settings from another machine.

- **Saving a workspace**—As you size, add, or remove windows in the Replication Console for Linux, you can save the workspace to use later or use on another Carbonite Availability client machine. Select **File** and one of the following options.
- **Save Workspace**—Save the current workspace. If you have not previously saved this workspace, you must specify a name for this workspace.
- **Save Workspace As**—Prompt for a new name when saving the current workspace.
- **Opening a workspace**—From the Replication Console for Linux, you can open a new workspace or open a previously saved workspace. Select **File** and one of the following options.
- **New Workspace**—Open an untitled workspace with the default Carbonite Availability window settings.
- **Open Workspace**—Open a previously saved workspace.

Clearing stored security credentials

Use the following steps to remove credentials cached in the Replication Console for Linux.

1. To access the credentials security option, select **File, Options** and select the **Security** tab.



2. To remove the security credentials, click **Clear Cached Security Credentials**.
3. Click **OK**.

Server settings

Most of the Carbonite Availability server settings are located in the Replication Console for Linux Server Properties dialog box. To access this dialog box, right-click a server in the left pane of the Replication Console and select **Properties**. The Server Properties dialog box contains multiple tabs with the Carbonite Availability server settings. For information on the server settings not available through the Replication Console for Linux, see the *DTCL Scripting Guide*.

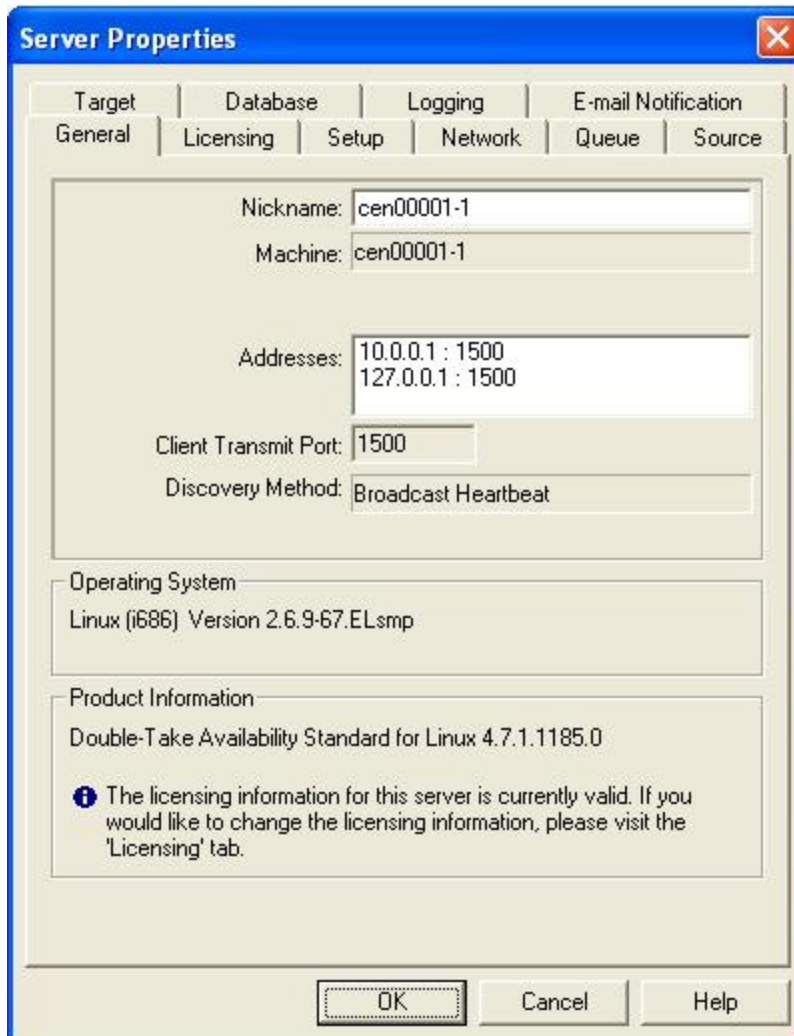
This section contains the following topics, each corresponding to a tab in the Server Properties dialog box.

- *Identifying a server* on page 26
- *Licensing a server* on page 28
- *Configuring server startup options* on page 31
- *Configuring network communication properties for a server* on page 33
- *Queuing data* on page 35
- *Configuring source data processing options* on page 38
- *Configuring target data processing options* on page 40
- *Specifying the Carbonite Availability database storage files* on page 41
- *Specifying file names for logging and statistics* on page 42
- *E-mailing system messages* on page 108

Identifying a server

From the Replication Console for Linux, you can see server identity information.

1. Right-click a server on the left pane of the Replication Console for Linux.
2. Select **Properties**
3. Select the **General** tab.



4. Specify the server identity information. Some of the fields are informational only.
 - **Nickname**—A nickname is saved in the Replication Console for Linux workspace, therefore, it only appears in the Replication Console for Linux on this server. It is not communicated across the network. If you export a workspace and use it on another Carbonite Availability server, the server nickname will appear there also.
 - **Machine**—This is the actual server name. This field is not modifiable.
 - **Addresses**—The IP address(es) for this server are listed in this field. This information is not modifiable and is displayed for your information. The machine's primary address is listed first.

- **Client Transmit Port**—This field displays the port that the Replication Console for Linux uses to send commands to a server. This port cannot be modified.
 - **Discovery Method**—This field indicates the method in which the Replication Console for Linux identifies the Carbonite Availability server.
 - **Manual**—A Carbonite Availability server was manually inserted into the Replication Console for Linux server tree.
 - **Broadcast Heartbeat**—A Carbonite Availability server is broadcasting Carbonite Availability heartbeats.
 - **Operating System**—The server's operating system version is displayed.
 - **Product Information**—The Carbonite Availability version number and build number are displayed.
5. Click **OK** to save the settings.

Licensing a server

From the Replication Console for Linux, you can manage your license keys. The license key is the Carbonite Availability license which is required on every Carbonite Availability server. The license key is a 24 character, alpha-numeric key. You can change your license key without reinstalling, if your license changes. There are different licenses available.

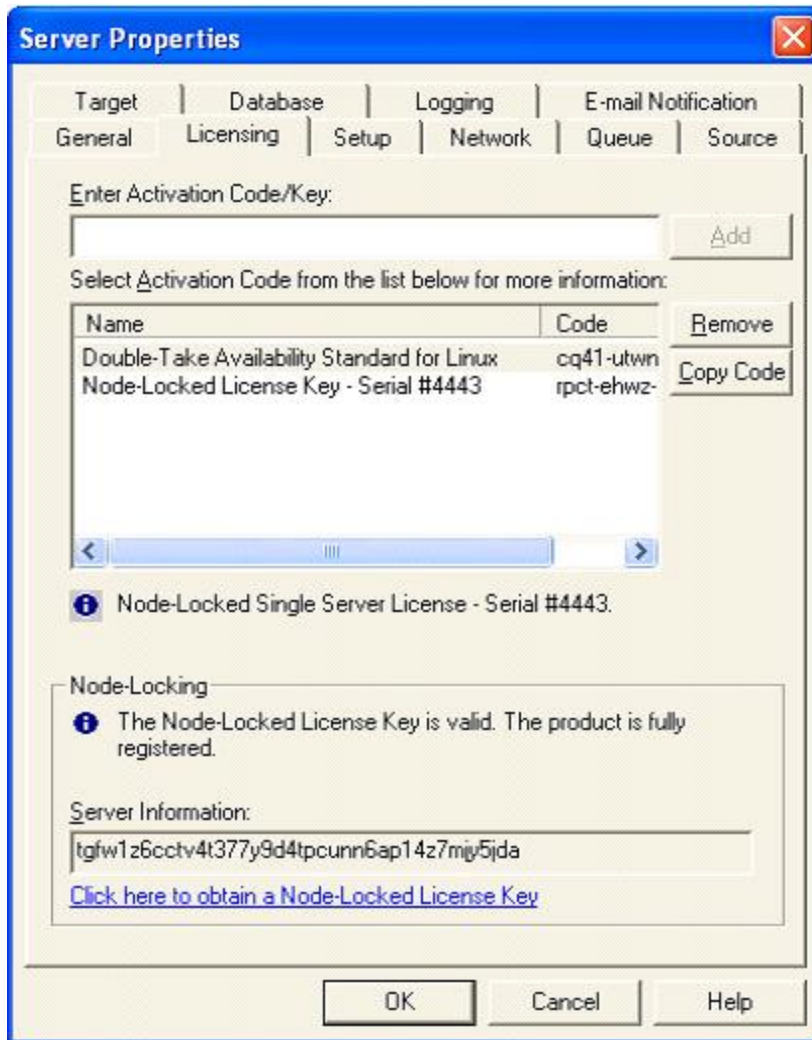
- **Evaluation**—An evaluation license has an expiration date built into the license key. When the license expires, the software will no longer function. The same evaluation licenses can be used on multiple machines on a network.
- **Single**—A single license is available on a per-machine basis. Each server is required to have a unique license whether it is functioning as a source, target, or both. A single license can only be used on one server on a network.
- **Site**—A site license is available to register every machine with the same license. This license is designed to be used on multiple servers on a network.

To prevent Carbonite Availability from being used illegally on multiple servers, you may have received a license key that must be activated from the Replication Console for Linux. Once the license key is entered, you have 14 days to activate it. The activation key can be obtained by supplying unique server information to Carbonite. Since the activation key contains unique server information, specific to the hardware where Carbonite Availability is installed, the activation key cannot be used on any other server, thus prohibiting illegal applications.

1. Right-click a server on the left pane of the Replication Console for Linux.
2. Select **Properties**.
3. Select the **Licensing** tab. The fields displayed on this tab will vary depending on your license keys.



The Replication Console for Linux Licensing tab uses older terminology, such as activation code and node-locking. The activation code is actually your license key before it is activated. Your node-locked code is the activation key that will activate your license.



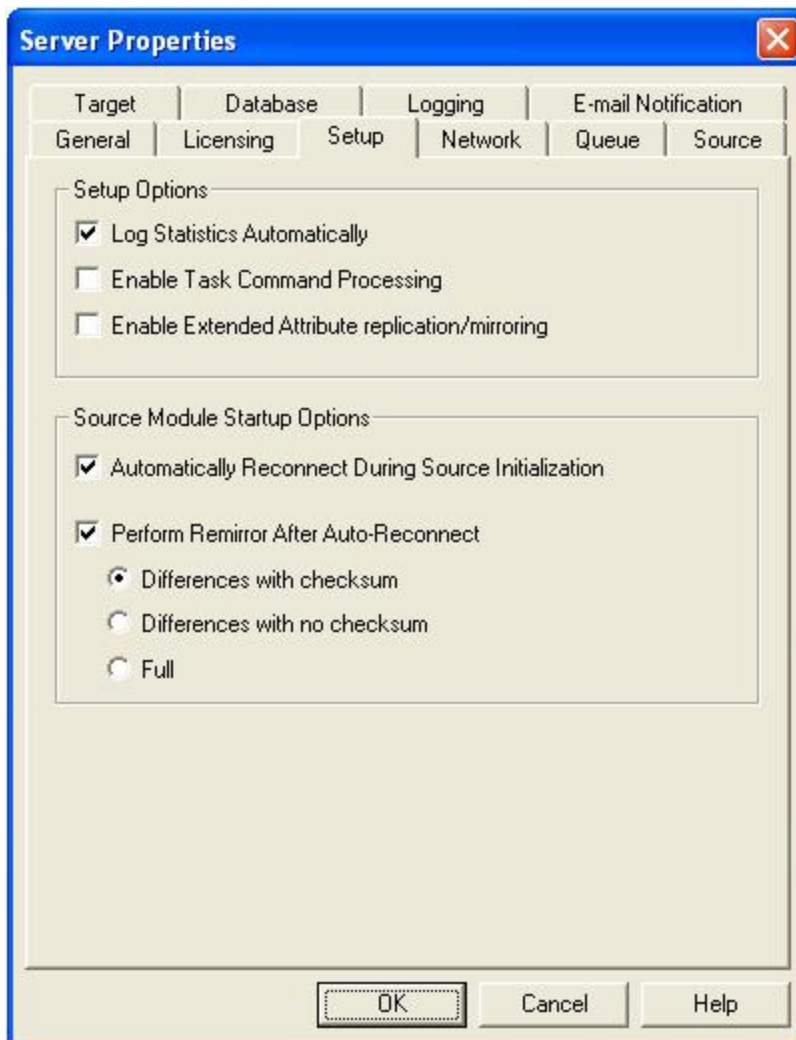
4. Enter a license key and click **Add**. Repeat for each license key.
5. Highlight an license key in the list to display any status messages for that key below the list display.
6. If you need to remove a key from the server, highlight it in the list and click **Remove**.
7. To activate a license key, you need to provide server information which will be used to generate an activation key.
 - a. After entering your license key, click **OK** to begin the grace period. At this point, you have 14 days to activate it.
 - b. Reopen the Server Properties **Licensing** tab.
 - c. Highlight your license key in the list to display the Node-Locking section at the bottom of the **Licensing** tab.
 - d. Click the hyperlink in the Node-Locking section. If you do not have an Internet connection, copy the **Server Information** text from the Node-Locking section into the form at <https://activate.doubletake.com> from another machine.
 - e. After you submit the form, you will receive an email with an activation key. Enter that key on the **Licensing** tab and click **Add**. The activation key is specific to this server. It cannot be used on any other server. If the activation key and server do not match, Carbonite Availability will not run.

8. Click **OK** to apply the keys you entered.

Configuring server startup options

From the Replication Console for Linux, you can configure server startup options for each Carbonite Availability server.

1. Right-click a server on the left pane of the Replication Console for Linux.
2. Select **Properties**
3. Select the **Setup** tab.



4. Specify the server setup and source startup options.
 - **Log Statistics Automatically**—If enabled, Carbonite Availability statistics logging will start automatically when Carbonite Availability is started.
 - **Enable Task Command Processing**—Task command processing is a Carbonite Availability feature that allows you to insert and run tasks at various points during the replication of data. Because the tasks are user-defined, you can achieve a wide variety of goals with this feature. For example, you might insert a task to create a snapshot or run a backup on the target after a certain segment of data from the source has been applied on

the target. This allows you to coordinate a point-in-time backup with real-time replication.

Task command processing can be enabled from the Replication Console for Linux, but it can only be initiated through the scripting language. See the *Scripting Guide* for more information.

If you disable this option on a source server, you can still submit tasks to be processed on a target, although task command processing must be enabled on the target.

- **Enable Extended Attribute replication/mirroring**—This option is no longer used.
- **Automatically Reconnect During Source Initialization**—If enabled, Carbonite Availability will automatically reconnect any connections that it automatically disconnected.
- **Perform Remirror After Auto-reconnect**—If enabled, Carbonite Availability will automatically perform a remirror after an auto-reconnect has occurred. You will also need to specify the type of mirror that you wish to perform after an auto-reconnect.
 - **Differences with Checksum**—Any file that is different on the source and target based on date, time, and/or size is flagged as different. The mirror then performs a checksum comparison on the flagged files and only sends those blocks that are different.
 - **Differences with no Checksum**—Any file that is different on the source and target based on date, time, and/or size is sent to the target.
 - **Full**—All files are sent to the target.

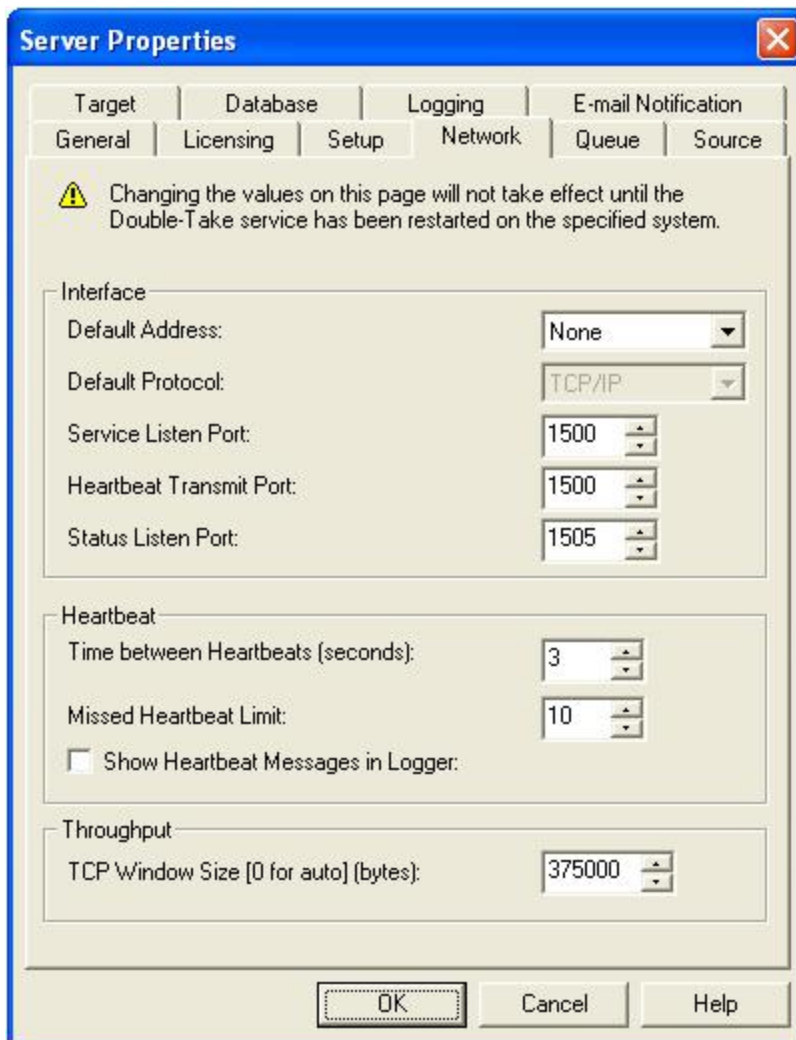


Database applications may update files without changing the date, time, or file size. Therefore, if you are using database applications, you should use the **Differences with checksum** or **Full** option.

5. Click **OK** to save the settings.

Configuring network communication properties for a server

1. Right-click a server on the left pane of the Replication Console for Linux.
2. Select **Properties**
3. Select the **Network** tab.



4. Specify the network communication properties.
 - **Default Address**—On a machine with multiple NICs, you can specify which address Carbonite Availability traffic will use. It can also be used on machines with multiple IP addresses on a single NIC.
 - **Default Protocol**—The default protocol for all Carbonite Availability communications is the TCP/IP protocol. In the future, Carbonite Availability may support other communication protocols.
 - **Service Listen Port**—Carbonite Availability servers use the **Service Listen Port** to send and receive commands and operations between two Carbonite Availability servers.
 - **Heartbeat Transmit Port**—A Carbonite Availability server sends its heartbeats to the **Heartbeat Transmit Port**.

- **Status Listen Port**—Carbonite Availability servers use the **Status Listen Port** to listen for requests from the Replication Console for Linux and other clients.
- **Time Between Heartbeats**—All Carbonite Availability servers transmit a heartbeat. This heartbeat allows other Carbonite Availability servers and Carbonite Availability clients to locate and identify the Carbonite Availability servers. The heartbeat is a broadcast UDP transmission. This heartbeat can be disabled, but if it is, Carbonite Availability will not auto-detect the Carbonite Availability servers to populate the Replication Console for Linux. By default, there are 3 seconds between heartbeats. If you set this option to 0, the heartbeats are disabled.
- **Missed Heartbeat Limit**—This is the number of heartbeats which can be missed before transmission is stopped and data is queued on the source.
- **Show Heartbeat Messages in Logger**—This checkbox enables the heartbeat messages in the Carbonite Availability log. Enabling this option will cause your logs to fill up faster.
- **TCP Window Size**—This option is the size, in bytes, of the buffer used for TCP transfers. This is an operating system buffer, not a Carbonite Availability buffer. If this option is set to zero (0), Linux kernel versions 2.6.7 or later can automatically tune this buffer setting for best server performance. Therefore, the recommended setting is 0 for automatic tuning, if you are using a version 2.6.7 or later Linux kernel. If you want to reduce or control network traffic, you can configure this option to a static size. The default is 375000 for a 1 GB network. Modifications should be relative to that speed using the calculation $37500 * \text{network_speed_in_bits_per_second} / 100 \text{ Mbit}$.



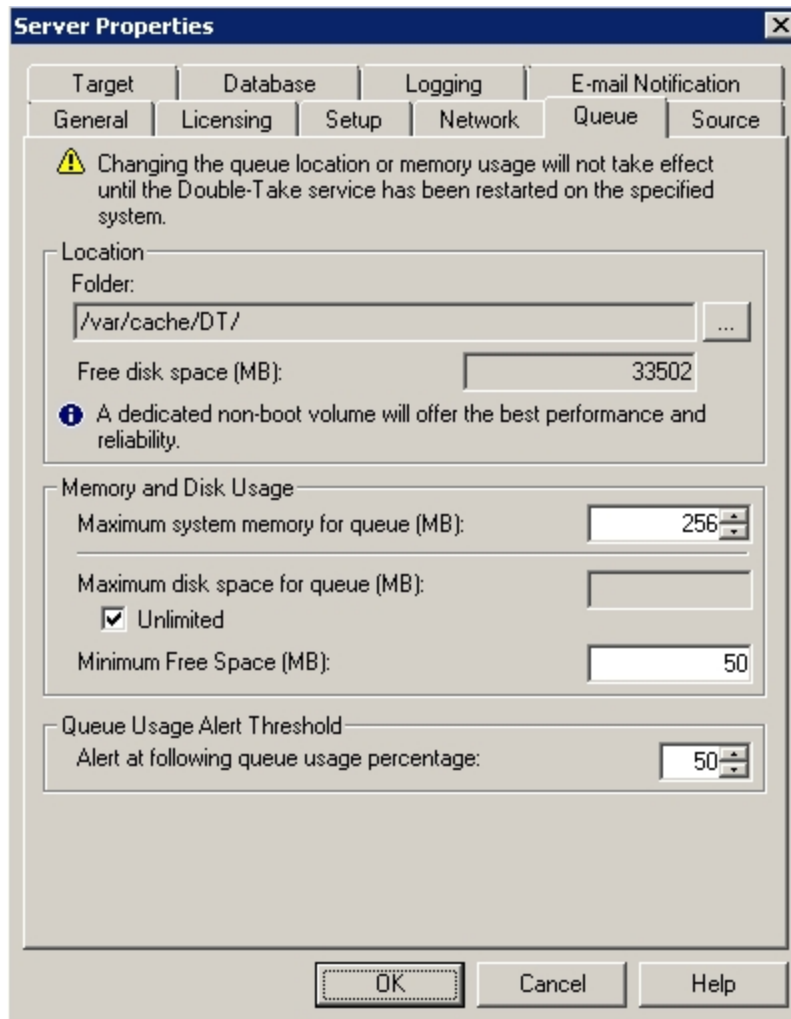
If you want to control network traffic, you may find the Carbonite Availability bandwidth limiting features to be a better method.

5. Click **OK** to save the settings.

Queuing data

You should configure queuing on both the source and target.

1. Right-click the server on the left pane of the Replication Console.
2. Select **Properties**.
3. Select the **Queue** tab.
4. Specify the queue settings for the server.



- **Folder**—This is the location where the disk queue will be stored. Carbonite Availability displays the amount of free space on the volume selected. Any changes made to the queue location will not take effect until the Double-Take service has been restarted on the server.

Select a location on a volume that will have minimal impact on the operating system and applications being protected. For best results and reliability, this should be a dedicated, non-boot volume. The disk queue should not be on the same physical or logical volume as the data being replicated.



Scanning the Carbonite Availability queue files for viruses can cause unexpected results. If anti-virus software detects a virus in a queue file and deletes or moves it, data integrity on the target cannot be guaranteed. As long as you have your anti-virus software configured to protect the actual production data, the anti-virus software can clean, delete, or move an infected file and the clean, delete, or move will be replicated to the target. This will keep the target from becoming infected and will not impact the Carbonite Availability queues.

- **Maximum system memory for queue**—This is the amount of system memory, in MB, that will be used to store data in queues. When exceeded, queuing to disk will be triggered. This value is dependent on the amount of physical memory available but has a minimum of 32 MB. By default, 128 MB of memory is used. If you set it lower, Carbonite Availability will use less system memory, but you will queue to disk sooner which may impact system performance. If you set it higher, Carbonite Availability will maximize system performance by not queuing to disk as soon, but the system may have to swap the memory to disk if the system memory is not available.

Since the source is typically running a production application, it is important that the amount of memory Carbonite Availability and the other applications use does not exceed the amount of RAM in the system. If the applications are configured to use more memory than there is RAM, the system will begin to swap pages of memory to disk and the system performance will degrade. For example, by default an application may be configured to use all of the available system memory when needed, and this may happen during high-load operations. These high-load operations cause Carbonite Availability to need memory to queue the data being changed by the application. In this case, you would need to configure the applications so that they collectively do not exceed the amount of RAM on the server. Perhaps on a server with 1 GB of RAM running the application and Carbonite Availability, you might configure the application to use 512 MB and Carbonite Availability to use 256 MB, leaving 256 MB for the operating system and other applications on the system. Many server applications default to using all available system memory, so it is important to check and configure applications appropriately, particularly on high-capacity servers.

Any changes to the memory usage will not take effect until the Double-Take service has been restarted on the server.

- **Maximum disk space for queue**—This is the maximum amount of disk space, in MB, in the specified **Folder** that can be used for Carbonite Availability disk queuing, or you can select **Unlimited** which will allow the queue usage to automatically expand whenever the available disk space expands. When the disk space limit is reached, Carbonite Availability will automatically begin the auto-disconnect process. By default, Carbonite Availability will use an unlimited amount of disk space. Setting this value to zero (0) disables disk queuing.
 - **Minimum Free Space**—This is the minimum amount of disk space in the specified **Folder** that must be available at all times. By default, 50 MB of disk space will always remain free. The **Minimum Free Space** should be less than the amount of physical disk space minus **Maximum disk space for queue**.
-



The **Maximum disk space for queue** and **Minimum Free Space** settings work in conjunction with each other. For example, assume your queues are stored on a

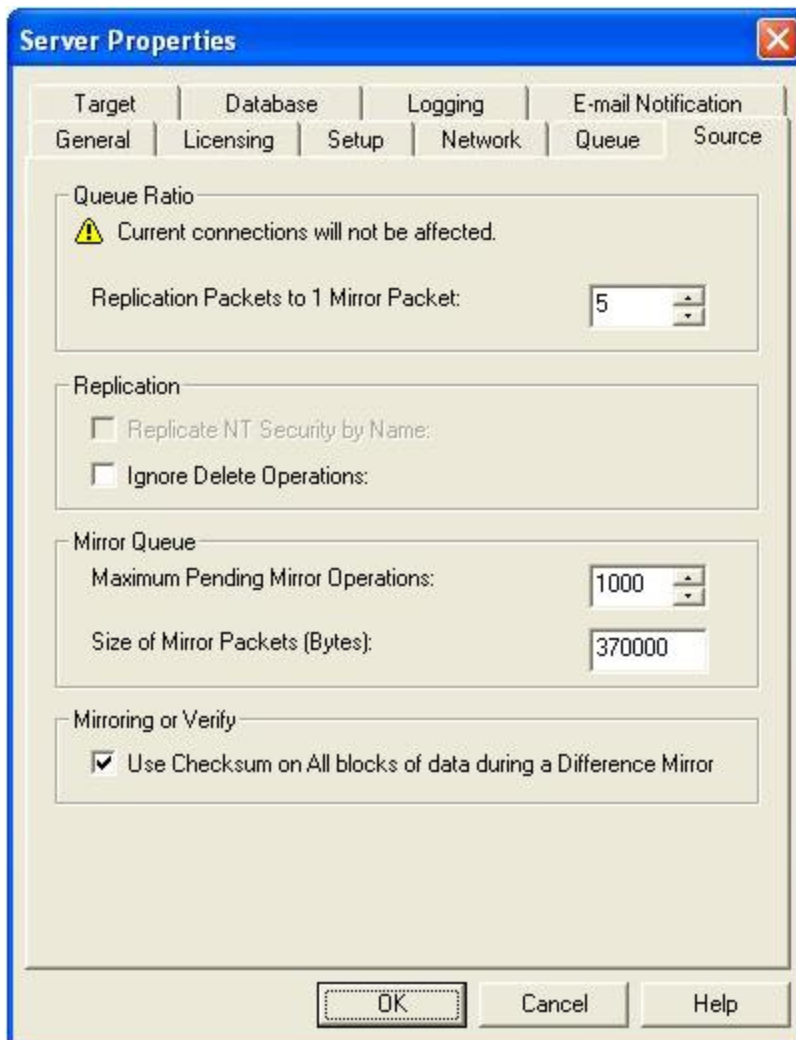


10 GB disk with the **Maximum disk space** for queue set to 10 GB and the **Minimum Free Space** set to 500 MB. If another program uses 5 GB, Carbonite Availability will only be able to use 4.5 GB so that 500 MB remains free.

- **Alert at following queue usage percentage**—This is the percentage of the disk queue that must be in use to trigger an alert message in the Carbonite Availability log. By default, the alert will be generated when the queue reaches 50%.
5. Click **OK** to save the settings.

Configuring source data processing options

1. Right-click a server on the left pane of the Replication Console for Linux.
2. Select **Properties**
3. Select the **Source** tab.



4. Specify how the source will process data.
 - **Replication Packets to 1 Mirror Packet**—You can specify the ratio of replication packets to mirror packets that are placed in the source queue. Specify a larger number if you have a busy network that has heavy replication. Also, if you anticipate increased network activity during a mirror, increase this number so that the replication queue does not get too large.
 - **Replicate NT Security by Name**—This is a Windows option only.
 - **Ignore Delete Operations**—This option allows you to keep files on the target machine after they are deleted on the source. When a file is deleted on the source, that delete operation is not sent to the target. (All edits to files on the source are still replicated to the

target; only deletions of whole files are ignored.) This option may be useful to give you an opportunity to make a backup of these files in the event they are needed in the future.



If delete operations are ignored long enough, the potential exists for the target to run out of space. In that case, you can manually delete files from the target to free space.

- **Maximum Pending Mirror Operations**—This option is the maximum number of mirror operations that are queued on the source. The default setting is 1000. If, during mirroring, the mirror queued statistic regularly shows low numbers, for example, less than 50, this value can be increased to allow Carbonite Availability to queue more data for transfer.
 - **Size of Mirror Packets**—This option determines the size of the mirror packets that Carbonite Availability transmits. The default setting is 32768 bytes.
 - **Use Checksum on All blocks of data during a Difference Mirror**—This option allows a file difference mirror to check each block of data, regardless of the file attributes. If this option is not marked, Carbonite Availability will assume files are synchronized if their attributes match.
-

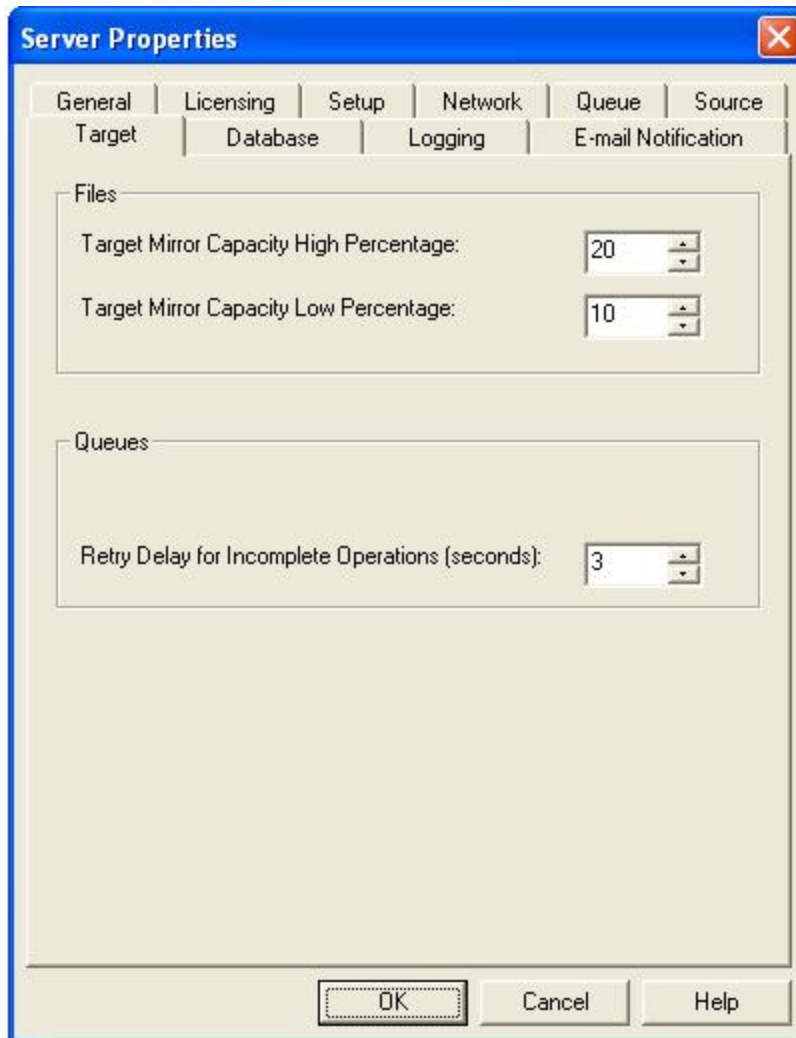


Database applications may update files without changing the date, time, or file size. Therefore, if you are using database applications, you should use the Block Checksum All option to ensure proper file comparisons.

5. Click **OK** to save the settings.

Configuring target data processing options

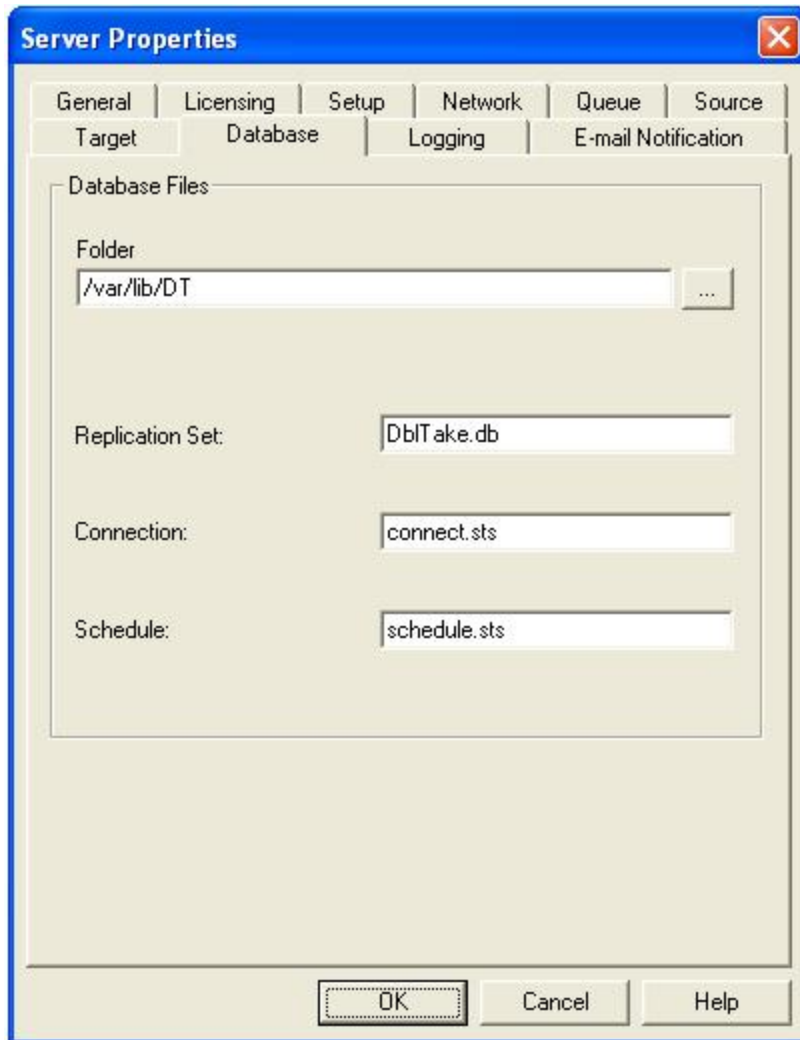
1. Right-click a server on the left pane of the Replication Console for Linux.
2. Select **Properties**
3. Select the **Target** tab.



4. Specify how the target will process data.
 - **Target Mirror Capacity High Percentage**—You can specify the maximum percentage of system memory that can contain mirror data before the target signals the source to pause the sending of mirror operations. The default setting is 20.
 - **Target Mirror Capacity Low Percentage**—You can specify the minimum percentage of system memory that can contain mirror data before the target signals the source to resume the sending of mirror operations. The default setting is 10.
 - **Retry Delay for Incomplete Operations (seconds)**—This option specifies the amount of time, in seconds, before retrying a failed operation on the target. The default setting is 3.
5. Click **OK** to save the settings.

Specifying the Carbonite Availability database storage files

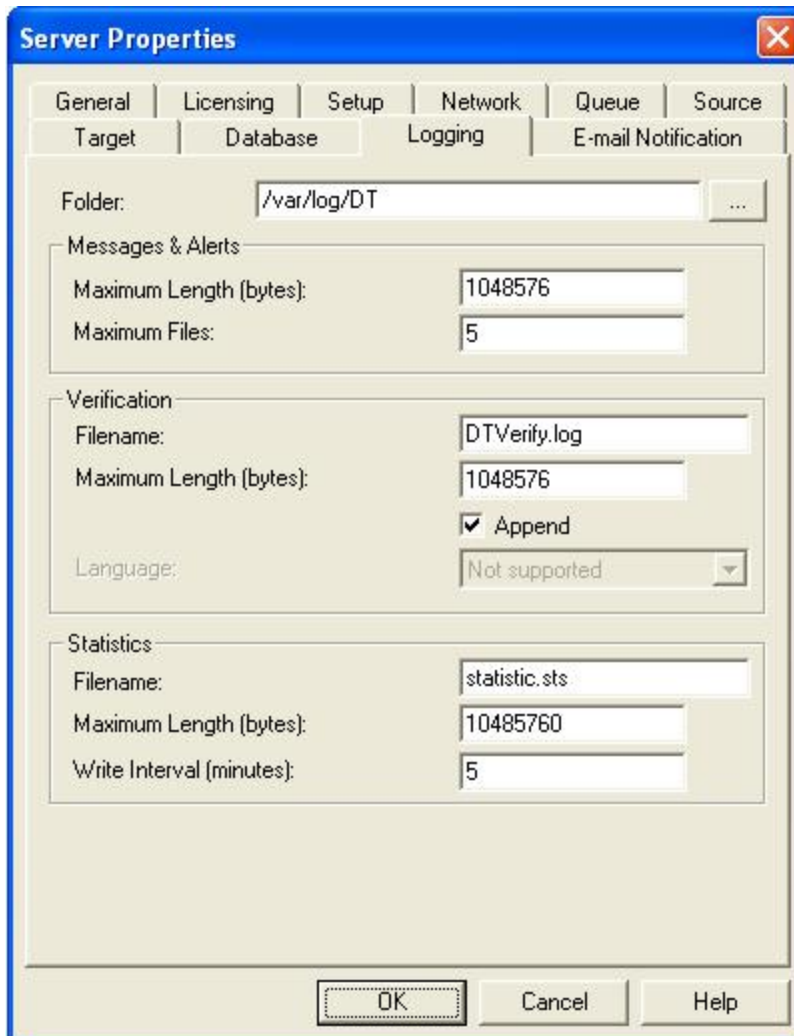
1. Right-click a server on the left pane of the Replication Console for Linux.
2. Select **Properties**
3. Select the **Database** tab.



4. Specify the database files that store the Carbonite Availability replication set, connection, and scheduling information.
 - **Folder**—Specify the directory where each of the database files on this tab are stored. The default location is the directory where the Carbonite Availability program files are installed.
 - **Replication Set**—This database file maintains which replication sets have been created on the server along with their names, rules, and so on. The default file name is DbITake.db.
 - **Connection**—This database file maintains the active source/target connection information. The default file name is connect.sts.
 - **Schedule**—This database file maintains any scheduling and transmission limiting options. The default file name is schedule.sts.
5. Click **OK** to save the settings.

Specifying file names for logging and statistics

1. Right-click a server on the left pane of the Replication Console for Linux.
2. Select **Properties**
3. Select the **Logging** tab.



4. Specify the location and file names for the log and statistics files.
 - **Folder**—Specify the directory where each of the log files on this tab are stored. The default location is the directory where the Carbonite Availability program files are installed.
 - **Messages & Alerts**
 - **Maximum Length**—Specify the maximum length of the client and service log files. The default size is 1048576 bytes and is limited by the available hard drive space.
 - **Maximum Files**—Specify the maximum number of Carbonite Availability alert log files that are maintained. The default is 5, and the maximum is 999.
 - **Verification**
 - **Filename**—The verification log is created during the verification process and details which files were verified as well as the files that are synchronized. This field contains

the name of the verification log, which is by default DTVerify.log.

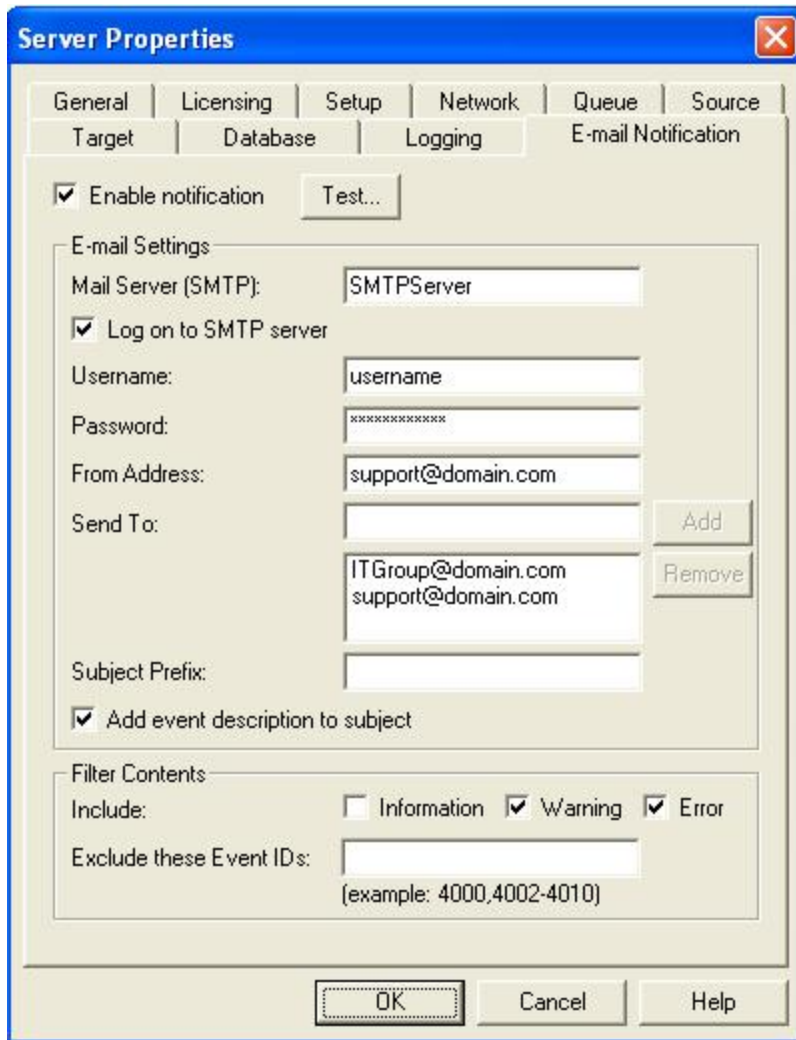
- **Maximum Length**—Specify the maximum length of the verification log file. The default maximum length is 1048576 bytes (1 MB).
- **Append**—Mark the Append check box if you want to append each verification process to the same log file. If this check box is not marked, each verification process that is logged will overwrite the previous log file. By default, this check box is selected.
- **Language**—At this time, English is the only language available.
- **Statistics**
 - **Filename**—The statistics log maintains connection statistics such as mirror bytes in queue or replication bytes sent. The default file name is statistic.sts. This file is a binary file that is read by the DTStat utility.
 - **Maximum Length**—Specify the maximum length of the statistics log file. The default maximum length is 10485760 bytes (10 MB). Once this maximum has been reached, Carbonite Availability begins overwriting the oldest data in the file.
 - **Write Interval**—Specify how often Carbonite Availability writes to the statistics log file. The default is every 5 minutes.

5. Click **OK** to save the settings.

E-mailing system messages

You can e-mail system messages to specified addresses. The subject of the e-mail will contain an optional prefix, the server name where the message was logged, the message ID, and the severity level (information, warning, or error). The text of the message will be displayed in the body of the e-mail message.

1. To enable e-mail notification for a server, right-click the server in the left pane of the Replication Console and select **Properties**.
2. Select the **E-mail Notification** tab.



The screenshot shows the 'Server Properties' dialog box with the 'E-mail Notification' tab selected. The 'Enable notification' checkbox is checked, and a 'Test...' button is visible. The 'E-mail Settings' section includes fields for 'Mail Server (SMTP): SMTPServer', 'Log on to SMTP server' (checked), 'Username: username', 'Password: [masked]', 'From Address: support@domain.com', and 'Send To:' (with 'Add' and 'Remove' buttons). The 'Subject Prefix' field is empty, and the 'Add event description to subject' checkbox is checked. The 'Filter Contents' section has 'Include:' checkboxes for 'Information' (unchecked), 'Warning' (checked), and 'Error' (checked). The 'Exclude these Event IDs:' field is empty, with an example '(example: 4000,4002-4010)' below it. The 'OK', 'Cancel', and 'Help' buttons are at the bottom.

3. Select **Enable notification**.



Any specified notification settings are retained when **Enable notification** is disabled.

4. Specify your e-mail settings.

- **Mail Server (SMTP)**—Specify the name of your SMTP mail server.
-



Specifying an SMTP server is the preferred method because it provides a direct connection between the mail server and Carbonite Availability, which decreases message latency and allows for better logging when the mail server cannot be reached.

If you do not specify an SMTP server, Carbonite Availability will attempt to use the Linux mail command. The success will depend on how the local mail system is configured. Carbonite Availability will be able to reach any address that the mail command can reach.

- **Log on to SMTP Server**—If your SMTP server requires authentication, enable **Log on to SMTP Server** and specify the **Username** and **Password** to be used for authentication. Your SMTP server must support the LOGIN authentication method to use this feature. If your server supports a different authentication method or does not support authentication, you may need to add the Carbonite Availability server as an authorized host for relaying e-mail messages. This option is not necessary if you are sending exclusively to e-mail addresses that the SMTP server is responsible for.
- **From Address**—Specify the e-mail address that you want to appear in the From field of each Carbonite Availability e-mail message. The address is limited to 256 characters.
- **Send To**—Specify the e-mail address that each Carbonite Availability e-mail message should be sent to and click **Add**. The e-mail address will be inserted into the list of addresses. Each address is limited to 256 characters. You can add up to 256 e-mail addresses. If you want to remove an address from the list, highlight the address and click **Remove**. You can also select multiple addresses to remove by Ctrl-clicking.
- **Subject Prefix** and **Add event description to subject**—The subject of each e-mail notification will be in the format Subject Prefix : Server Name : Message Severity : Message ID : Message Description. The first and last components (Subject Prefix and Message Description) are optional. The subject line is limited to 150 characters.

If desired, enter unique text for the **Subject Prefix** which will be inserted at the front of the subject line for each Carbonite Availability e-mail message. This will help distinguish Carbonite Availability messages from other messages. This field is optional.

If desired, enable **Add event description** to subject to have the description of the message appended to the end of the subject line. This field is optional.

- **Filter Contents**—Specify which messages that you want to be sent via e-mail. Specify **Information**, **Warning**, and/or **Error**. You can also specify which messages to exclude based on the message ID. Enter the message IDs as a comma or semicolon separated list. You can indicate ranges within the list.
-



You can test e-mail notification by specifying the options on the E-mail Notification tab and clicking **Test**. If desired, you can send the test message to a different e-mail address by selecting **Send To** and entering a comma or semicolon separated list of addresses. Modify the message text up to 1024 characters, if necessary. Click **Send** to test the e-mail notification. The results will be displayed in a message box.

Click **OK** to close the message and click **Close** to return to the E-mail Notification tab. 

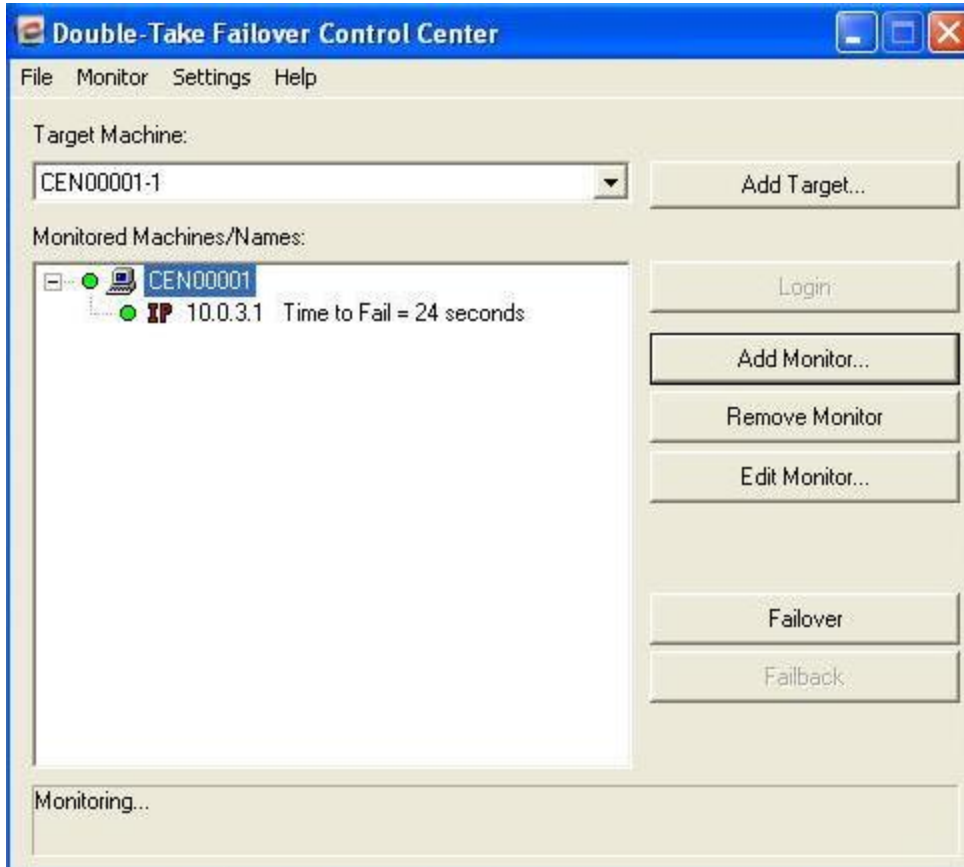
If an error occurs while sending an e-mail, a message will be generated. This message will not trigger an e-mail. Subsequent e-mail errors will not generate additional messages. When an e-mail is sent successfully, a message will then be generated. If another e-mail fails, one message will again be generated. This is a cyclical process where one message will be generated for each group of failed e-mail messages, one for each group of successful e-mail messages, one for the next group of failed messages, and so on.

If you start and then immediately stop the Double-Take service, you may not get e-mail notifications for the log entries that occur during startup.

By default, most virus scan software blocks unknown processes from sending traffic on port 25. You need to modify the blocking rule so that Carbonite Availability e-mail messages are not blocked.

Failover for Linux console for files and folders jobs

From the Failover for Linux console, you can manage, monitor, and control failover for your Carbonite Availability servers. The Failover for Linux console displays a main window for monitoring failover activity. Control buttons to the right allow you to configure and manage your servers.



Setting the frequency of Failover for Linux console refreshes

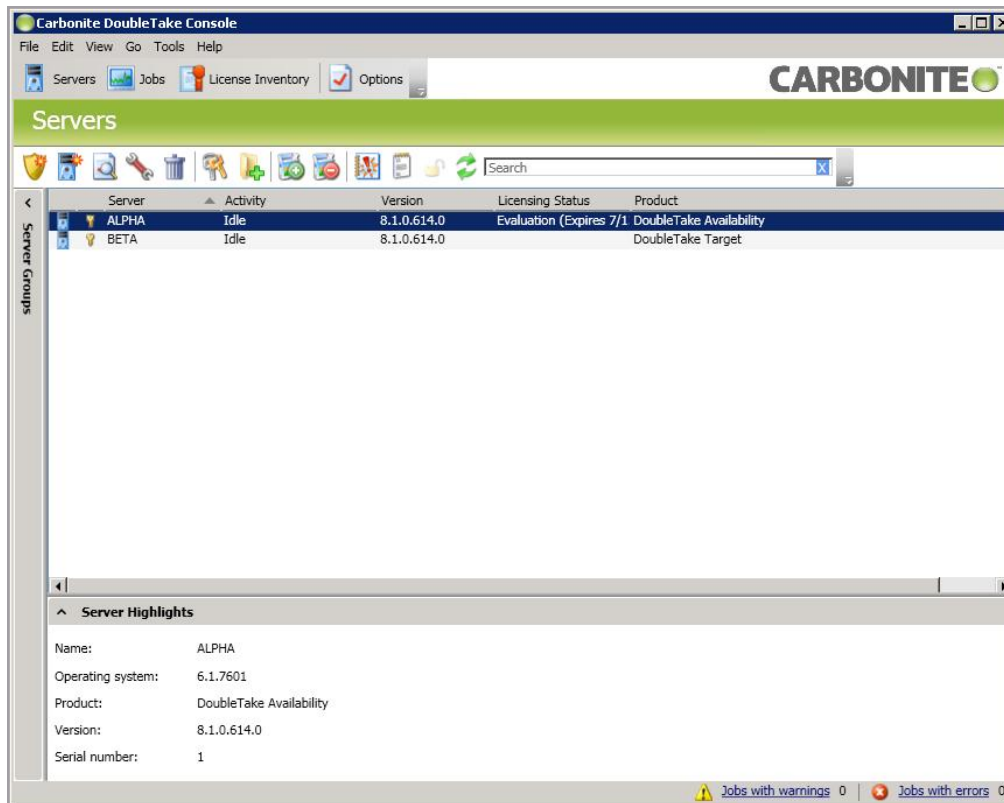
The failover client periodically requests information from the source and target. Depending on the type of information, the request may be a machine-specific request, like obtaining the **Time to Fail** status from a target, or may be a general request, like determining which machines are running Carbonite Availability.

The rate at which these requests are made can be modified through the Failover for Linux console refresh rate dialog box. Select **Settings, Refresh Rate**. The default update interval is one second. A lower refresh rate value updates the information in the Failover for Linux console window's **Monitored Machines** tree more often, but also generates more network traffic and higher utilization on the client and target machines. A higher refresh rate value updates the information less frequently, but minimizes the network traffic.

Carbonite Replication Console for full server and full server to ESX jobs

After you have installed the console, you can launch it by selecting **Carbonite, Replication, Carbonite Replication Console** from your **Programs, All Programs, or Apps**, depending on your operating system.

The Carbonite Replication Console is used to protect and monitor your servers and jobs. Each time you open the Carbonite Replication Console, you start at the **Servers** page which allows you to view, edit, add, remove, or manage the servers in your console. You can also create a new job from this page.



At the bottom of the Carbonite Replication Console, you will see a status bar. At the right side, you will find links for **Jobs with warnings** and **Jobs with errors**. This lets you see quickly, no matter which page of the console you are on, if you have any jobs that need your attention. Select this link to go to the **Jobs** page, where the appropriate **Filter: Jobs with warnings** or **Filter: Jobs with errors** will automatically be applied.



The first time you start the console, you will see the getting started screen tips on the **Servers** page. These tips walk you through the basic steps of adding a server to your console, installing Carbonite Availability on that server, and creating a job on that server. If you do not want to see the tips, close them. If you want to reopen the tips after you have closed them, select **Help, Show Getting Started Tips**.

You can manually check for Carbonite Availability updates by selecting **Help, Check for Updates**.

- **Update available**—If there is an update available, click **Get Update**. The dialog box will close and your web browser will open to the Carbonite web site where you can download and install the update.
 - **No update available**—If you are using the most recent console software, that will be indicated. Click **Close**.
 - **No connection available**—If the console cannot contact the update server or if there is an error, the console will report that information. The console log contains a more detailed explanation of the error. Click **Check using Browser** if you want to open your browser to check for console software updates. You will need to use your browser if your Internet access is through a proxy server.
-

Carbonite Replication Console requirements

You must meet the following requirements for the Carbonite Replication Console.

- **Operating system**—The Carbonite Replication Console can be run from a Windows source or target. It can also be run from a physical or virtual machine running Windows 10, Windows 8, or Windows 7 Service Pack 1 or later.
- **Microsoft .NET Framework**—Microsoft .NET Framework version 4.5.1 is required.
- **Screen resolution**—For best results, use a 1024x768 or higher screen resolution.



The Carbonite Availability installation prohibits the console from being installed on Server Core. Because Windows 2012 allows you to switch back and forth between Server Core and a full installation, you may have the console files available on Server Core, if you installed Carbonite Availability while running in full operating system mode. In any case, you cannot run the Carbonite Replication Console on Server Core.

Console options

There are several options that you can set that are specific to the Carbonite Replication Console. To access these console options, select **Options** from the toolbar.

- **Monitoring**—This section is used to determine how the console monitors your Carbonite Availability servers.
 - **Monitoring interval**—Specifies how often, in seconds, the console refreshes the monitoring data. The servers will be polled at the specified interval for information to refresh the console.
 - **Automatic retry**—This option will have the console automatically retry server login credentials, after the specified retry interval, if the server login credentials are not accepted. Keep in mind the following caveats when using this option.
 - This is only for server credentials, not job credentials.
 - A set of credentials provided for or used by multiple servers will not be retried for the specified retry interval on any server if it fails on any of the servers using it.
 - Verify your environment's security policy when using this option. Check your policies for failed login lock outs and resets. For example, if your policy is to reset the failed login attempt count after 30 minutes, set this auto-retry option to the same or a slightly larger value as the 30 minute security policy to decrease the chance of a lockout.
 - Restarting the Carbonite Replication Console will automatically initiate an immediate login.
 - Entering new credentials will initiate an immediate login using the new credentials.
 - **Retry on this interval**—If you have enabled the automatic retry, specify the length of time, in minutes, to retry the login.
- **Server Communication**—This section is used to determine how the console communicates with your Carbonite Availability servers.
 - **Default port for XML web services protocol**—Specifies the port that the console will use when sending and receiving data to Carbonite Availability servers. By default, the port is 6325. Changes to the console port will not take effect until the console is restarted.
 - **Default port for legacy protocol**—If you are using an older Carbonite Availability version, you will need to use the legacy protocol port. This applies to Carbonite Availability versions 5.1 or earlier.
- **Diagnostics**—This section assists with console troubleshooting.
 - **Export Diagnostic Data**—This button creates a raw data file that can be used for debugging errors in the Carbonite Replication Console. Use this button as directed by technical support.
 - **View Log File**—This button opens the Carbonite Replication Console log file. Use this button as directed by technical support. You can also select **View, View Console Log File** to open the Carbonite Replication Console log file.
 - **View Data File**—This button opens the Carbonite Replication Console data file. Use this button as directed by technical support. You can also select **View, View Console Data File** to open the Carbonite Replication Console data file.
- **License Inventory**—This section controls if the console contains a license inventory. This

feature may not appear in your console if your service provider has restricted access to it.

- **Enable license inventory**—This option allows you to use this console to manage the Carbonite Availability licenses assigned to your organization. When this option is enabled, the **License Inventory** page is also enabled.
- **Default Installation Options**—All of the fields under the **Default Installation Options** section are used by the push installation on the **Install** page. The values specified here will be the default options used for the push installation.
 - **Activate online after install completes**—Specify if you want to activate your Carbonite Availability licenses at the end of the installation. The activation requires Internet access from the console machine or the machine you are installing to. Activation will be attempted from the console machine first and if that fails, it will be attempted from the machine you are installing to. If you choose not to have the installation activate your licenses, you will have to activate them through the console license inventory or the server's properties page.
 - **Location of install folders**—Specify the parent directory location where the installation files are located. The parent directory can be local on your console machine or a UNC path.
 - **Windows**—Specify the parent directory where the Windows installation file is located. The default location is where the Carbonite Replication Console is installed, which is `\Program Files\Carbonite\Replication`. The console will automatically use the `\x64` subdirectory which is populated with the Windows installation files when you installed the console. If you want to use a different location, you must copy the `\x64` folder and its installation file to the different parent directory that you specify.
 - **Linux**—For Linux servers, you have two choices.
 - If you copied the Linux installation files from your download to your Carbonite Replication Console installation location, you must make sure they are in a `\Linux` subdirectory under the parent directory you specified for **Location of install folders**. Copy the Linux `.deb` or `.rpm` files from your download to the `\Linux` subdirectory. Make sure you only have a single version of the Linux installation files in that location. The push installation cannot determine which version to install if there are multiple versions in the `\Linux` subdirectory.
 - If you have already deployed your Linux virtual recovery appliance, specify the UNC path to the installers share on the appliance. For example, if your appliance is called `DTAppliance`, use the path `\\DTAppliance\opt\dbtk\share\installers` for the **Location of install folders**. The console will automatically use the installation files in the `\linux` subdirectory of this share location. (This option is only available if you deployed an appliance from an `.ova` file. There are no installers available if you installed Carbonite Availability on your appliance manually.)
- **Default Windows Installation Options**—All of the fields under the **Default Installation Options** section are used by the push installation on the **Install** page. The values specified here will be the default options used for the push installation.
 - **Temporary folder for installation package**—Specify a temporary location on the server where you are installing Carbonite Availability where the installation files will be copied and run.
 - **Installation folder**—Specify the location where you want to install Carbonite Availability on each server. This field is not used if you are upgrading an existing version of Carbonite Availability. In that case, the existing installation folder will be used.

- **Queue folder**—Specify the location where you want to store the Carbonite Availability disk queue on each server.
 - **Amount of system memory to use**—Specify the maximum amount of memory, in MB, that can be used for Carbonite Availability processing.
 - **Minimum free disk space**—This is the minimum amount of disk space in the specified **Queue folder** that must be available at all times. This amount should be less than the amount of physical disk space minus the disk size specified for **Limit disk space for queue**.
 - **Do not use disk queue**—This option will disable disk queuing. When system memory has been exhausted, Carbonite Availability will automatically begin the auto-disconnect process.
 - **Unlimited disk queue**—Carbonite Availability will use an unlimited amount of disk space in the specified **Queue folder** for disk queuing, which will allow the queue usage to automatically expand whenever the available disk space expands. When the available disk space has been used, Carbonite Availability will automatically begin the auto-disconnect process.
 - **Limit disk space for queue**—This option will allow you to specify a fixed amount of disk space, in MB, in the specified **Queue folder** that can be used for Carbonite Availability disk queuing. When the disk space limit is reached, Carbonite Availability will automatically begin the auto-disconnect process.
-



If the servers you are pushing to do not have a C drive, make sure you update the folder fields because the Carbonite Replication Console will not validate that the fields are set to a volume that does not exist and the installation will not start.

- **Default Linux Installation Options**—All of the fields under the **Default Installation Options** section are used by the push installation on the **Install** page. The values specified here will be the default options used for the push installation.
 - **Temporary folder for installation package**—Specify a temporary location on the server where you are installing Carbonite Availability where the installation files will be copied and run.

Managing servers

To manage the servers in your console, select **Servers** from the toolbar. The **Servers** page is for server management and job creation.

- **Add and remove servers**—You can add servers to and remove servers from the console.
- **View and edit**—You can view server details and edit Carbonite Availability server properties.
- **Create job**—You can create a protection or migration job for a selected server.
- **Server organization**—You can organize the servers that are in your console into groups, allowing you to filter the servers you are viewing based on your organization.

Review the following sections to understand the information and controls available on the **Servers** page.



If you have uninstalled and reinstalled Carbonite Availability on a server, you may see the server twice on the **Servers** page because the reinstall assigns a new unique identifier to the server. One of the servers (the original version) will show with the red X icon. You can safely remove that server from the console.

Left pane

You can expand or collapse the left pane by clicking on the **Server Highlights** heading. This pane allows you to organize your servers into folders. The servers displayed in the top right pane will change depending on the server group folder selected in the left pane. Every server in your console session is displayed when the **All Servers** group is selected. If you have created and populated server groups under **My Servers**, then only the servers in the selected group will be displayed in the right pane.

Between the main toolbar and the left pane is a smaller toolbar. These toolbar options control the server groups in the left pane.

Create New Server Group

Creates a new server group below the selected group

Rename Server Group

Allows you to rename the selected server group

Delete Server Group

Deletes the selected server group. This will not delete the servers in the group, only the group itself.

Overflow Chevron








Displays any toolbar buttons that are hidden from view when the window size is reduced.

Top right pane

The top pane displays high-level overview information about your servers. You can sort the data within a column in ascending and descending order. You can also move the columns to the left or right of each other to create your desired column order. The list below shows the columns in their default left to right order.





Column 1 (Blank)

The first blank column indicates the machine type.

-  Carbonite Availability source or target server which could be a physical server, virtual machine, or a cluster node
-  Carbonite Availability source or target server which is a Windows cluster
-  vCenter server
-  ESX server
-  Carbonite Availability Reporting Service server
-  Offline server which means the console cannot communicate with this machine.
-  Any server icon with a red circle with white X overlay is an error which means the console can communicate with the machine, but it cannot communicate with Carbonite Availability on it.

Column 2 (Blank)

The second blank column indicates the security level

-  Processing—The console is attempting to communicate with machine.
-  Administrator access—This level grants full control.
-  Monitor only access—This level grants monitoring privileges only.
-  No security access—This level does not allow monitoring or control.

Server

The name or IP address of the server. If you have specified a reserved IP address, it will be displayed in parenthesis.

Activity

There are many different **Activity** messages that keep you informed of the server activity. Most of the activity messages are informational and do not require any administrator interaction. If you see error messages, check the server details. See *Viewing server details* on page 67.

Version

The Carbonite Availability product version information, if any.

Licensing Status

The status of the license, if any, on the server. If your license is expired, any jobs using that server will be in an error state. If you have multiple licenses, the status will indicate the license that requires the soonest action. For example, if you have a Carbonite Move license that expires in two days and a Carbonite Availability license that must be activated within 10 days, the status will be for the Carbonite Move license.

Product

The Carbonite Availability products, if any, licensed for the server

Bottom right pane

The details displayed in the bottom pane provide additional information for the server highlighted in the top pane. You can expand or collapse the bottom pane by clicking on the **Server Highlights** heading.

Name

The name or IP address of the server.

Operating system

The operating system of the server. This field will not be displayed if the console cannot connect to Carbonite Availability on the server.

Product

The Carbonite Availability products, if any, licensed for the server

Version

The product version information, if any

Serial Number

The serial number associated with the Carbonite Availability license

Toolbar

The following options are available on the main toolbar of the **Servers** page. Some options are only available for a single selected server and others are available for multiple selected servers.

Create a New Job

The available job creation choices depend on the Carbonite Availability licenses applied to your server.

- **Protect**—If you are licensed for Carbonite Availability, use the **Protect** option to create a protection job for the selected server.
- **Migrate**—If you are licensed for Carbonite Move or certain Carbonite Availability licenses, use the **Migrate** option to create a migration job for the selected server.

Add Servers

Adds a new server. This button leaves the **Servers** page and opens the **Add Servers** page. See *Adding servers* on page 64.

View Server Details

Views detailed information about a server. This button leaves the **Servers** page and opens the **View Server Details** page. See *Viewing server details* on page 67.

Edit Server Properties

Edits the server's properties and options. This button leaves the **Servers** page and opens the **Edit Server Properties** page. See *Editing server properties* on page 69.

Remove Server

Removes the server from the console.

Provide Credentials

Changes the login credentials that the Carbonite Replication Console use to authenticate to a server. This button opens the **Provide Credentials** dialog box where you can specify the new account information. See *Providing server credentials* on page 66. You will remain on the **Servers** page after updating the server credentials.

Manage Group Assignments

Allows you to assign, move, and remove the selected server from specific server groups. This buttons opens the Manage Group Assignments dialog box where you can

assign and unassign the server to specific server groups. The server will appear in server groups marked with a checkmark, and will not appear in groups without a checkmark. Servers assigned to a server group will automatically appear in parent server groups.

Install

Installs or upgrades Carbonite Availability on the selected server. This button opens the **Install** page where you can specify installation options.

Uninstall

Uninstalls Carbonite Availability on the selected server.

View Server Events

Views Windows application event messages for a server. This option is not available for Linux sources or appliances.

View Server Logs

Views the Carbonite Availability logs messages for a server. This button opens the **Logs** window. This separate window allows you to continue working in the Carbonite Replication Console while monitoring log messages. You can open multiple logging windows for multiple servers. When the Carbonite Replication Console is closed, all logging windows will automatically close.

Activate Online

Activates licenses and applies the activation keys to servers in one step. You must have Internet access for this process. You will not be able to activate a license that has already been activated.

Refresh

Refreshes the status of the selected servers.

Search

Allows you to search the product or server name for items in the list that match the criteria you have entered.

Overflow Chevron

Displays any toolbar buttons that are hidden from view when the window size is reduced.

Right-click menu

The following options are available on the right-click menu of the **Servers** page. Some options are only available for a single selected server and others are available for multiple selected servers.

Protect

If you are licensed for Carbonite Availability, use the **Protect** option to create a protection job for the selected server.

Migrate

If you are licensed for Carbonite Move or certain Carbonite Availability licenses, use the **Migrate** option to create a migration job for the selected server.

View Server Details

Views detailed information about a server. This button leaves the **Servers** page and opens the **View Server Details** page. See *Viewing server details* on page 67.

Edit Server Properties

Edits the server's properties and options. This button leaves the **Servers** page and opens the **Edit Server Properties** page. See *Editing server properties* on page 69.

Remove Server

Removes the server from the console.

Provide Credentials

Changes the login credentials that the Carbonite Replication Console use to authenticate to a server. This button opens the **Provide Credentials** dialog box where you can specify the new account information. See *Providing server credentials* on page 66. You will remain on the **Servers** page after updating the server credentials.

Manage Group Assignments

Allows you to assign, move, and remove the selected server from specific server groups. This buttons opens the Manage Group Assignments dialog box where you can assign and unassign the server to specific server groups. The server will appear in server groups marked with a checkmark, and will not appear in groups without a checkmark. Servers assigned to a server group will automatically appear in parent server groups.

Install

Installs or upgrades Carbonite Availability on the selected server. This button opens the **Install** page where you can specify installation options.

Uninstall

Uninstalls Carbonite Availability on the selected server.

Copy

Copies the information for the selected servers. You can then paste the server information as needed. Each server is pasted on a new line, with the server information being comma-separated.

Paste

Pastes a new-line separated list of servers into the console. Your copied list of servers must be entered on individual lines with only server names or IP addresses on each line.

View Server Events

Views Windows event messages for a server. This option is not available for Linux sources or appliances.

View Server Logs

Views the Carbonite Availability logs messages for a server. This button opens the **Logs** window. This separate window allows you to continue working in the Carbonite Replication Console while monitoring log messages. You can open multiple logging windows for multiple servers. When the Carbonite Replication Console is closed, all logging windows will automatically close.

Activate Online

Activates licenses and applies the activation keys to servers in one step. You must have Internet access for this process. You will not be able to activate a license that has already been activated.

Gather Support Diagnostics

Executes the diagnostic DTInfo utility which collects configuration data for use when reporting problems to technical support. It gathers Carbonite Availability log files; Carbonite Availability and system settings; network configuration information such as IP, WINS, and DNS addresses; and other data which may be necessary for technical support to troubleshoot issues. You will be prompted for a location to save the resulting file which is created with the information gathered. Because this utility is gathering several pieces of information, across the network to your console machine, it may take

several minutes to complete the information gathering and sending the resulting file to the console machine.

View Replication Service Details 

Views the replication service details for a server. This option is not applicable to Linux source servers or appliances.

Refresh 

Refreshes the status of the selected servers.

Adding servers

The first time you start the console, the **Servers** page is empty. In order to protect and monitor your servers, you must insert your servers and/or appliances in the console.

Inserting servers manually

1. Click **Servers** from the main toolbar.
2. Click **Add servers** from the **Servers** page toolbar.
3. Select **Add servers** and click **Next**.
 - **Server**—This is the name or IP address of the server or appliance to be added to the console.



If you enter the source server's fully-qualified domain name, the Carbonite Replication Console will resolve the entry to the server short name. If that short name resides in two different domains, this could result in name resolution issues. In this case, enter the IP address of the server.

If you are using a NAT environment, make sure you add your server to the Carbonite Replication Console using the correct public or private IP address. The name or IP address you use to add a server to the console is dependent on where you are running the console. Specify the private IP address of any servers on the same side of the router as the console. Specify the public IP address of any servers on the other side of the router as the console.

-
- **User name**—Specify a local user that is a member of the **dtadmin** or **dtmon** security group on the server.
 - **Password**—Specify the password associated with the **User name** you entered.
 - **Domain**—If you are working in a domain environment, specify the **Domain**.
 - **Management Service port**—If you want to change the port used by the Double-Take Management Service, disable **Use default port** and specify the port number you want to use. This option is useful in a NAT environment where the console needs to be able to communicate with the server using a specific port number. Use the public or private port depending on where the console is running in relation to the server you are adding.
4. After you have specified the server or appliance information, click **Add**.
 5. Repeat steps 3 and 4 for any other servers or appliances you want to add.
 6. If you need to remove servers or appliances from the list of **Servers to be added**, highlight a server and click **Remove**. You can also remove all of them with the **Remove All** button.
 7. When your list of **Servers to be added** is complete, click **OK**.

Importing and exporting servers from a server and group configuration file

You can share the console server and group configuration between machines that have the Carbonite Replication Console installed. The console server configuration includes the server group configuration, server name, server communications ports, and other internal processing information.

To export a server and group configuration file, select **File, Export Servers**. Specify a file name and click **Save**. After the configuration file is exported, you can import it to another console.

When you are importing a console server and group configuration file from another console, you will not lose or overwrite any servers that already exist in the console. For example, if you have server alpha in your console and you insert a server configuration file that contains servers alpha and beta, only the server beta will be inserted. Existing group names will not be merged, so you may see duplicate server groups that you will have to manually update as desired.

To import a server and group configuration file, select **File, Import Servers**. Locate the console configuration file saved from the other machine and click **Open**.

Providing server credentials

To update the security credentials used for a specific server, select **Provide Credentials** from the toolbar on the **Servers** page. When prompted, specify the **User name**, **Password**, and **Domain** of the account you want to use for this server. Click **OK** to save the changes.

Viewing server details

Highlight a server on the **Servers** page and click **View Server Details** from the toolbar. The **View Server Details** page allows you to view details about that particular server. The server details vary depending on the type of server or appliance you are viewing.

Server name

The name or IP address of the server. If you have specified a reserved IP address, it will be displayed in parenthesis.

Operating system

The server's operating system version

Roles

The role of this server in your Carbonite Availability environment. In some cases, a server can have more than one role.

- **Engine Role**—Source or target server
- **Reporting Service**—Reporting Service server

Status

There are many different **Status** messages that keep you informed of the server activity. Most of the status messages are informational and do not require any administrator interaction. If you see error messages, check the rest of the server details.

Activity

There are many different **Activity** messages that keep you informed of the server activity. Most of the activity messages are informational and do not require any administrator interaction. If you see error messages, check the rest of the server details.

Connected via

The IP address and port the server is using for communications. You will also see the Carbonite Availability protocol being used to communicate with server. The protocol will be XML web services protocol (for servers running Carbonite Availability version 5.2 or later) or Legacy protocol (for servers running version 5.1 or earlier).

Version

The product version information

Access

The security level granted to the specified user

User name

The user account used to access the server

Licensing

Licensing information for the server

Source jobs

A list of any jobs from this server. Double-clicking on a job in this list will automatically open the **View Job Details** page.

Target jobs

A list of any jobs to this server. Double-clicking on a job in this list will automatically open the **View Job Details** page.

Editing server properties

Right-click a server on the **Servers** page and select **Edit server properties**. The **Edit Server Properties** page allows you to view and edit properties for that server. Click on a heading on the **Edit Server Properties** page to expand or collapse a section of properties.

- *General server properties* on page 70—Encryption configuration
- *Server licensing* on page 71—Views, adds, and removes license keys
- *E-mail notification configuration* on page 73—Configures e-mail notification

General server properties

The general server properties allow you to enable or disable encryption. Use this option to encrypt your data before it is sent from the source to the target. Both the source and target must be encryption capable (version 8.0.0 or later), however this option only needs to be enabled on the source or target in order to encrypt data. Keep in mind that all jobs from a source with this option enabled or to a target with this option enabled will have the same encryption setting. Changing this option will cause jobs to auto-reconnect and possibly remirror. The encryption method used is AES-256.



Server licensing

Licensing identifies your Carbonite Availability license keys.



The fields and buttons in the **Licensing** section will vary depending on your Carbonite Replication Console configuration and the type of license keys you are using.

Licensing

Add license keys and activation keys:

Current license keys:

	Product	Serial Number	Expiration Date
	Double-Take Availability for Windows...	7323	10/12/2014

Changes to licensing are applied immediately.

Activation

This server has one or more licenses that require activation. To fully activate your software you can use the activate-online feature to complete the activation process using your internet connection. Alternatively using the link below, you may obtain activation keys from our activation website and add them above to activate your software.

Server information:

[Get an activation key from our website](#)

- **Add license keys and activation keys**—Your license key or activation key is a 24 character, alpha-numeric key. You can change your license key without reinstalling, if your license changes. To add a license key or activation key, type in the key or click **Choose from inventory** and select a key from your console's license inventory. Then click **Add**.



The license inventory feature cannot be enabled if your service provider has restricted access to it.

- **Current license keys**—The server's current license key information is displayed. To remove a key, highlight it and click **Remove**. To copy a key, highlight it and click **Copy**. To replace a key,

enter a new key and click **Add**. If you are replacing an unexpired key with the same version and serial number, you should not have to reactivate it and any existing jobs will continue uninterrupted. If you are replacing an unexpired key with a new version or new serial number or replacing an expired key, you will have to reactivate and remirror.

- **Activation**—If your license key needs to be activated, you will see an additional **Activation** section at the bottom of the **Licensing** section. To activate your key, use one of the following procedures.
 - **Activate online**—If you have Internet access, you can activate your license and apply the activated license to the server in one step by selecting **Activate Online**.



You will not be able to activate a license that has already been activated.

- **Obtain activation key online, then activate**—If you have Internet access, click the hyperlink in the **Activation** section to take you to the web so that you can submit your activation information. Complete and submit the activation form, and you will receive an e-mail with the activation key. Activate your server by entering the activation key in the **Add license keys and activations keys** field and clicking **Add**.
- **Obtain activation key offline, then activate**—If you do not have Internet access, go to <https://activate.doubletake.com> from another machine that has Internet access. Complete and submit the activation form, and you will receive an e-mail with the activation key. Activate your server by entering the activation key in the **Add license keys and activations keys** field and clicking **Add**.

The activation key is specific to this server. It cannot be used on any other server. If the activation key and server do not match, Carbonite Availability will not run.

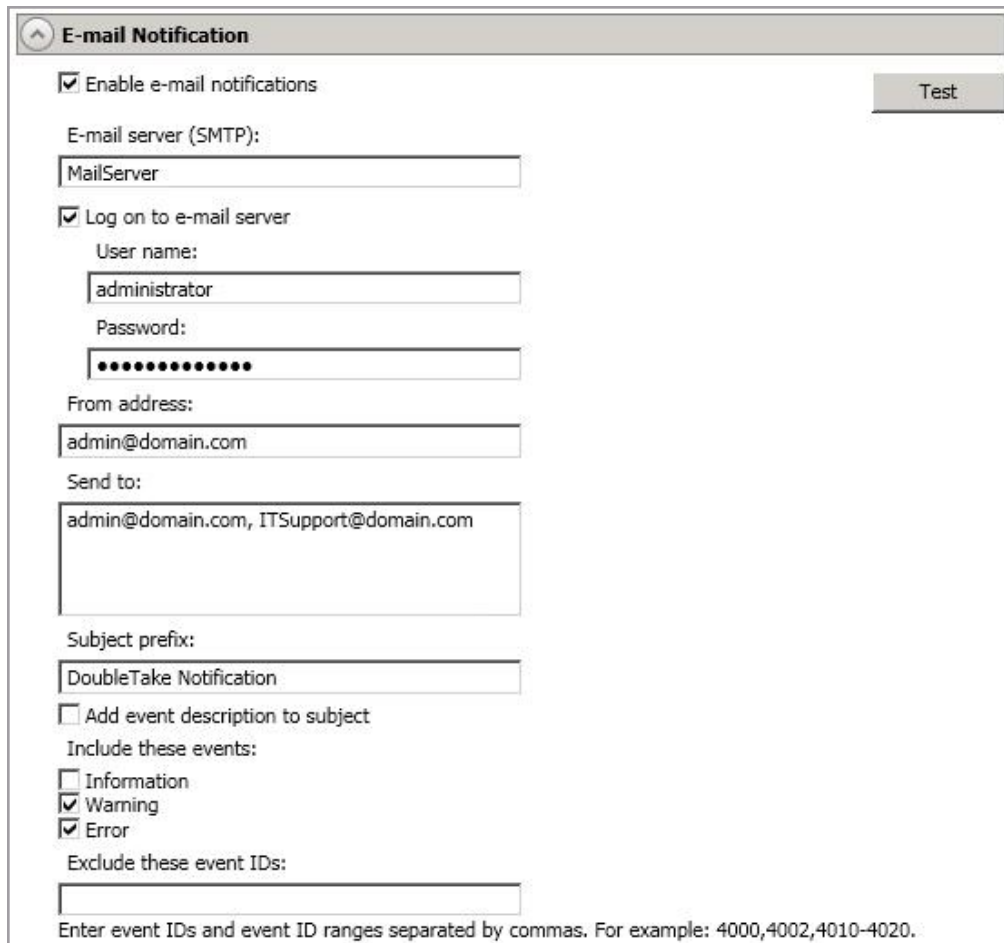


If your Carbonite Availability license keys needs to be activated, you will have 14 days to do so.

If you need to rename a server that already has a Carbonite Availability license applied to it, you should deactivate that license before changing the server name. That includes rebuilding a server or changing the case (capitalization) of the server name (upper or lower case or any combination of case). If you have already rebuilt the server or changed the server name or case, you will have to perform a host-transfer to continue using that license.

E-mail notification configuration

You can email Carbonite Availability event messages to specific addresses using an SMTP mail server. The subject of the e-mail will contain an optional prefix, the server name where the message was logged, the message ID, and the severity level (information, warning, or error). The text of the event message will be displayed in the body of the e-mail message.



The screenshot shows the 'E-mail Notification' configuration window. It includes a 'Test' button and several input fields and checkboxes. The 'Enable e-mail notifications' checkbox is checked. The 'E-mail server (SMTP)' field contains 'MailServer'. The 'Log on to e-mail server' checkbox is checked, with 'User name' set to 'administrator' and 'Password' masked with dots. The 'From address' field contains 'admin@domain.com'. The 'Send to' field contains 'admin@domain.com, ITSupport@domain.com'. The 'Subject prefix' field contains 'DoubleTake Notification'. The 'Add event description to subject' checkbox is unchecked. Under 'Include these events', 'Warning' and 'Error' are checked, while 'Information' is unchecked. The 'Exclude these event IDs' field is empty. A note at the bottom states: 'Enter event IDs and event ID ranges separated by commas. For example: 4000,4002,4010-4020.'

- **Enable e-mail notification**—This option enables the e-mail notification feature. Any specified notification settings will be retained if this option is disabled.
- **E-mail server**—Specify the name of your SMTP mail server.
- **Log on to e-mail server**—If your SMTP server requires authentication, enable this option and specify the **User name** and **Password** to be used for authentication. Your SMTP server must support the LOGIN authentication method to use this feature. If your server supports a different authentication method or does not support authentication, you may need to add the Carbonite Availability server as an authorized host for relaying e-mail messages. This option is not necessary if you are sending exclusively to e-mail addresses that the SMTP server is responsible for.
- **From address**—Specify the e-mail address that you want to appear in the From field of each Carbonite Availability e-mail message. The address is limited to 256 characters.

- **Send to**—Specify the e-mail addresses that each Carbonite Availability e-mail message should be sent to. Enter the addresses as a comma or semicolon separated list. Each address is limited to 256 characters. You can add up to 256 e-mail addresses.
- **Subject prefix** and **Add event description to subject**—The subject of each e-mail notification will be in the format Subject Prefix : Server Name : Message Severity : Message ID : Message Description. The first and last components (Subject Prefix and Message Description) are optional. The subject line is limited to 100 characters.

If desired, enter unique text for the **Subject prefix** which will be inserted at the front of the subject line for each Carbonite Availability e-mail message. This will help distinguish Carbonite Availability messages from other messages. This field is optional.

If desired, enable **Add event description to subject** to have the description of the message appended to the end of the subject line. This field is optional.

- **Includes these events**—Specify which messages that you want to be sent via e-mail. Specify **Information**, **Warning**, and/or **Error**. You can also specify which messages to exclude based on the message ID. Enter the message IDs as a comma or semicolon separated list. You can indicate ranges within the list.



When you modify your e-mail notification settings, you will receive a test e-mail summarizing your new settings. You can also test e-mail notification by clicking **Test**. By default, the test will be run from the machine where the console is running. If desired, you can send the test message to a different e-mail address by selecting **Send To** and entering a comma or semicolon separated list of addresses. Modify the **Message Text** up to 1024 characters, if necessary. Click **Send** to test the e-mail notification. The results will be displayed in a message box.

If an error occurs while sending an e-mail, a message will be generated. This message will not trigger another e-mail. Subsequent e-mail errors will not generate additional messages. When an e-mail is sent successfully, a message will then be generated. If another e-mail fails, one message will again be generated. This is a cyclical process where one message will be generated for each group of failed e-mail messages, one for each group of successful e-mail messages, one for the next group of failed messages, and so on.

If you start and then immediately stop the Double-Take service, you may not get e-mail notifications for the log entries that occur during startup.

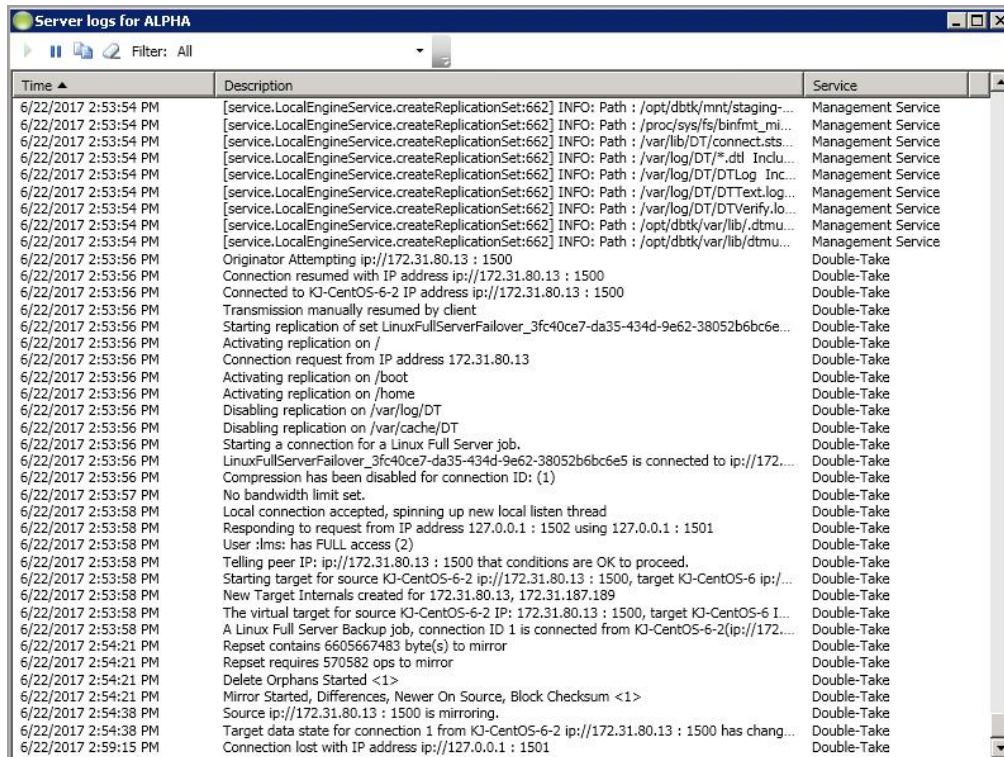
By default, most anti-virus software blocks unknown processes from sending traffic on port 25. You need to modify the blocking rule so that Carbonite Availability e-mail messages are not blocked.

Viewing server logs

You can view the engine and Management Service logs using either of these two methods.

- On the **Servers** page, highlight a server in the list and click **View Server Logs** from the toolbar.
- On the **Jobs** page, right-click a job and select **View Logs**. Select either the source server log or the target server log.

Separate logging windows allow you to continue working in the Carbonite Replication Console while monitoring log messages. You can open multiple logging windows for multiple servers. When the Carbonite Replication Console is closed, all logging windows will automatically close.



Time	Description	Service
6/22/2017 2:53:54 PM	[service.LocalEngineService.createReplicationSet:662] INFO: Path : /opt/dbtk/mnt/staging-...	Management Service
6/22/2017 2:53:54 PM	[service.LocalEngineService.createReplicationSet:662] INFO: Path : /proc/sys/fs/binfmt_mi...	Management Service
6/22/2017 2:53:54 PM	[service.LocalEngineService.createReplicationSet:662] INFO: Path : /var/lib/DT/connect.sts...	Management Service
6/22/2017 2:53:54 PM	[service.LocalEngineService.createReplicationSet:662] INFO: Path : /var/log/DT/*.*.dtl Inclu...	Management Service
6/22/2017 2:53:54 PM	[service.LocalEngineService.createReplicationSet:662] INFO: Path : /var/log/DT/DTLog Inc...	Management Service
6/22/2017 2:53:54 PM	[service.LocalEngineService.createReplicationSet:662] INFO: Path : /var/log/DT/DTText.log...	Management Service
6/22/2017 2:53:54 PM	[service.LocalEngineService.createReplicationSet:662] INFO: Path : /var/log/DT/DTVerify.lo...	Management Service
6/22/2017 2:53:54 PM	[service.LocalEngineService.createReplicationSet:662] INFO: Path : /opt/dbtk/var/lib/.dtmu...	Management Service
6/22/2017 2:53:54 PM	[service.LocalEngineService.createReplicationSet:662] INFO: Path : /opt/dbtk/var/lib/dtmu...	Management Service
6/22/2017 2:53:56 PM	Originator Attempting ip://172.31.80.13 : 1500	Double-Take
6/22/2017 2:53:56 PM	Connection resumed with IP address ip://172.31.80.13 : 1500	Double-Take
6/22/2017 2:53:56 PM	Connected to KJ-CentOS-6-2 IP address ip://172.31.80.13 : 1500	Double-Take
6/22/2017 2:53:56 PM	Transmission manually resumed by client	Double-Take
6/22/2017 2:53:56 PM	Starting replication of set LinuxFullServerFailover_3fc40ce7-da35-434d-9e62-38052b6bc6e...	Double-Take
6/22/2017 2:53:56 PM	Activating replication on /	Double-Take
6/22/2017 2:53:56 PM	Connection request from IP address 172.31.80.13	Double-Take
6/22/2017 2:53:56 PM	Activating replication on /boot	Double-Take
6/22/2017 2:53:56 PM	Activating replication on /home	Double-Take
6/22/2017 2:53:56 PM	Disabling replication on /var/log/DT	Double-Take
6/22/2017 2:53:56 PM	Disabling replication on /var/cache/DT	Double-Take
6/22/2017 2:53:56 PM	Starting a connection for a Linux Full Server job.	Double-Take
6/22/2017 2:53:56 PM	LinuxFullServerFailover_3fc40ce7-da35-434d-9e62-38052b6bc6e5 is connected to ip://172.31.80.13 : 1500	Double-Take
6/22/2017 2:53:56 PM	Compression has been disabled for connection ID: (1)	Double-Take
6/22/2017 2:53:57 PM	No bandwidth limit set.	Double-Take
6/22/2017 2:53:58 PM	Local connection accepted, spinning up new local listen thread	Double-Take
6/22/2017 2:53:58 PM	Responding to request from IP address 127.0.0.1 : 1502 using 127.0.0.1 : 1501	Double-Take
6/22/2017 2:53:58 PM	User :lms: has FULL access (2)	Double-Take
6/22/2017 2:53:58 PM	Telling peer IP: ip://172.31.80.13 : 1500 that conditions are OK to proceed.	Double-Take
6/22/2017 2:53:58 PM	Starting target for source KJ-CentOS-6-2 ip://172.31.80.13 : 1500, target KJ-CentOS-6 ip://172.31.80.13 : 1500	Double-Take
6/22/2017 2:53:58 PM	New Target Internals created for 172.31.80.13, 172.31.187.189	Double-Take
6/22/2017 2:53:58 PM	The virtual target for source KJ-CentOS-6-2 IP: 172.31.80.13 : 1500, target KJ-CentOS-6 I...	Double-Take
6/22/2017 2:53:58 PM	A Linux Full Server Backup job, connection ID 1 is connected from KJ-CentOS-6-2(ip://172.31.80.13 : 1500)	Double-Take
6/22/2017 2:54:21 PM	Repet contains 6605667483 byte(s) to mirror	Double-Take
6/22/2017 2:54:21 PM	Repet requires 570582 ops to mirror	Double-Take
6/22/2017 2:54:21 PM	Delete Orphans Started <1>	Double-Take
6/22/2017 2:54:21 PM	Mirror Started, Differences, Newer On Source, Block Checksum <1>	Double-Take
6/22/2017 2:54:38 PM	Source ip://172.31.80.13 : 1500 is mirroring.	Double-Take
6/22/2017 2:54:38 PM	Target data state for connection 1 from KJ-CentOS-6-2 ip://172.31.80.13 : 1500 has chang...	Double-Take
6/22/2017 2:59:15 PM	Connection lost with IP address ip://127.0.0.1 : 1501	Double-Take

The following table identifies the controls and the table columns in the **Server logs** window.

Start 

This button starts the addition and scrolling of new messages in the window.

Pause 

This button pauses the addition and scrolling of new messages in the window. This is only for the **Server logs** window. The messages are still logged to their respective files on the server.

Copy

This button copies the messages selected in the **Server logs** window to the Windows clipboard.

Clear

This button clears the **Server logs** window. The messages are not cleared from the respective files on the server. If you want to view all of the messages again, close and reopen the **Server logs** window.

Filter

From the drop-down list, you can select to view all log messages or only those messages from the Double-Take log or the Management Service log.

Time

This column in the table indicates the date and time when the message was logged.

Description

This column in the table displays the actual message that was logged.

Service

This column in the table indicates if the message is from the Double-Take log or the Management Service log.

Managing VMware servers

To manage your VMware servers, select **Go, Manage VMware Servers**. The **Manage VMware Server** page allows you to view, add, remove, or edit credentials for your VMware servers available in the console.

VMware Server

The name of the VMware server

Full Name

The full name of the VMware server

User Name

The user account being used to access the VMware server

Add VMware Server

Add a new VMware server. When prompted, specify the VMware server and a user account. If you are using a non-default port for your server, specify the server followed by a colon and then the port number, for example, 112.47.12.7:85. If your server name does not match the security certificate or the security certificate has expired, you will be prompted if you want to install the untrusted security certificate.

Remove Server

Remove the VMware server from the console.

Provide Credentials

Edit credentials for the selected VMware server. When prompted, specify a user account to access the VMware server.

Managing snapshots

Use the instructions below to manage the snapshots that Carbonite Availability has taken.

1. From the **Jobs** page, highlight the job and click **Manage Snapshots** in the toolbar.
2. You will see the list of snapshots, if any, associated with the job.
 - **Scheduled**—This snapshot was taken as part of a periodic snapshot.
 - **Deferred**—This snapshot was taken as part of a periodic snapshot, although it did not occur at the specified interval because the job between the source and target was not in a good state.
 - **Manual**—This snapshot was taken manually by a user.
 - **Test**—This snapshot was taken as part of a test failover.
3. Click **Take Snapshot** to create a new snapshot for the job.
4. If there is a snapshot that you no longer need, highlight it in the list and click **Delete**.
5. When you have completed your snapshot management, click **Close**.

Chapter 3 Files and folders protection

Create a files and folders job when you want to protect data. You can also use it to protect applications, such as Oracle or MySQL, however you will need to use your own customized failover and failback scripts to start and stop services during failover and failback. This job type does not protect a server's system state. Use the following links to access information and steps specific to files and folder protection.

- *Files and folders requirements* on page 80
- *Creating a files and folders job* on page 83
- *Protection monitoring* on page 96
- *Connections* on page 111
- *Mirroring* on page 114
- *Replication* on page 121
- *Verification* on page 138
- *Data transmission* on page 144
- *Failover* on page 154
- *Failback and restoration* on page 167

Files and folders requirements

Each Carbonite Availability server must meet minimum requirements. Verify that each server meets the requirements for the function of that machine. Additionally, the machine where you will be running the console must also meet some basic requirements.

- *Source and target server requirements* on page 80
- *Console requirements* on page 82

Source and target server requirements

- **Operating system**—Make sure your servers meet the operating system, kernel, and file system requirements.
 - **Operating system**—Red Hat Enterprise Linux, Oracle Enterprise Linux, and CentOS
 - **Version**—5.9 through 5.11
 - **Kernel type for x86 (32-bit) architectures**—Default, SMP, Xen, PAE
 - **Kernel type for x86-64 (64-bit) architectures**—Default, SMP, Xen
 - **File system**—Ext3, Ext4, XFS
 - **Notes**—Oracle Enterprise Linux support is for the mainline kernel only, not the Unbreakable kernel.
 - **Operating system**—Red Hat Enterprise Linux, Oracle Enterprise Linux, and CentOS
 - **Version**—6.7 through 6.9
 - **Kernel type for x86 (32-bit) architectures**—Default
 - **Kernel type for x86-64 (64-bit) architectures**—Default
 - **File system**—Ext3, Ext4, XFS (64-bit only)
 - **Operating system**—Red Hat Enterprise Linux, Oracle Enterprise Linux, and CentOS
 - **Version**—7.2 through 7.4
 - **Kernel type for x86 (32-bit) architectures**—No 32-bit architectures are supported
 - **Kernel type for x86-64 (64-bit) architectures**—Default
 - **File system**—Ext3, Ext4, XFS
 - **Notes**—Oracle Enterprise Linux version 7.4 is for the Mainline kernel only.
 - **Operating system**—SUSE Linux Enterprise
 - **Version**—11.2 through 11.4
 - **Kernel type for x86 (32-bit) architectures**—Default, Xen, XenPAE, VMI
 - **Kernel type for x86-64 (64-bit) architectures**—Default, Xen
 - **File system**—Ext3, XFS
 - **Operating system**—SUSE Linux Enterprise
 - **Version**—12.0 through 12.2
 - **Kernel type for x86 (32-bit) architectures**—No 32-bit architectures are supported

- **Kernel type for x86-64 (64-bit) architectures**—Default
- **File system**—Ext3, Ext4, XFS, Btrfs
- **Notes**—If you are planning to convert an existing file system to Btrfs, you must delete any existing Carbonite Availability jobs and re-create them after converting to Btrfs.
- **Operating system**—Ubuntu
 - **Version**—12.04.3, 12.04.4, and 12.04.5
 - **Kernel type for x86 (32-bit) architectures**—Generic
 - **Kernel type for x86-64 (64-bit) architectures**—Generic
 - **File system**—Ext2, Ext3, Ext4, XFS
- **Operating system**—Ubuntu
 - **Version**—14.04.3, 14.04.4, and 14.04.5
 - **Kernel type for x86 (32-bit) architectures**—Generic
 - **Kernel type for x86-64 (64-bit) architectures**—Generic
 - **File system**—Ext2, Ext3, Ext4, XFS
- **Operating system**—Ubuntu
 - **Version**—16.04.0, 16.04.1, and 16.04.2
 - **Kernel type for x86 (32-bit) architectures**—Generic
 - **Kernel type for x86-64 (64-bit) architectures**—Generic
 - **File system**—Ext2, Ext3, Ext4, XFS



For all operating systems except Ubuntu, the kernel version must match the expected kernel for the specified release version. For example, if `/etc/redhat-release` declares the system to be a Redhat 6.6 system, the kernel that is installed must match that.

Carbonite Availability does not support stacking filesystems, like eCryptFS.

You must have `sshd` (or the package that installs `sshd`), `lsb`, `parted`, `/usr/sbin/dmidecode`, and `/usr/bin/which` on your Linux servers before you can install and use Carbonite Availability. See your operating system documentation for details on these packages and utilities.

- **System Memory**—The minimum system memory on each server should be 1 GB. The recommended amount for each server is 2 GB.
- **Disk Usage**—The amount of disk space required for the Carbonite Availability program files is approximately 85 MB. About 45 MB will be located on your `/`(root) partition, and the remainder will be on your `/usr` partition. You will need to verify that you have additional disk space for Carbonite Availability queuing, logging, and so on. Additionally, on a target server, you need sufficient disk space to store the replicated data from all connected sources, allowing additional space for growth.
- **Protocols**—Your servers must have TCP/IP. IPv4 is the only supported version.
- **NAT support**—Carbonite Availability supports IP and port forwarding in NAT environments with the following caveats.
 - Only IPv4 is supported.
 - Only standalone servers are supported.

- Make sure you have added your server to the Carbonite Replication Console using the correct public or private IP address. The name or IP address you use to add a server to the console is dependent on where you are running the console. Specify the private IP address of any servers on the same side of the router as the console. Specify the public IP address of any servers on the other side of the router as the console.
- DNS failover and updates will depend on your configuration
 - Only the source or target can be behind a router, not both.
 - The DNS server must be routable from the target
- **Ports**—Port 1501 is used for localhost communication. Ports 1500, 1505, 1506, 6325, and 6326 are used for component communication and must be opened on any firewall that might be in use.
- **IP address and subnet configuration**—Because of limitations in the way the Linux kernel handles IP address aliases, do not mix subnets on the eth0 network interface. Failover should not cause problems in this configuration, but you will lose IP addresses during failback. Therefore, if you must mix subnets on a single interface, use eth1 or higher.
- **Name resolution**—Your servers must have name resolution or DNS. The Replication Console for Linux and interactive text client (DTCL -i) will fail if there is no DNS entry or way for a server to resolve server names. For details on name resolution options, see your Linux documentation or online Linux resources.
- **Security**—Carbonite Availability security is granted through membership in user groups. The groups can be local or LDAP (Lightweight Directory Access Protocol). A user must provide a valid local account that is a member of the Carbonite Availability security groups.
- **SELinux policy**—SELinux should be disabled on the source and target.
- **Docker**—Your source cannot be a Docker host.
- **VMware Tools**—Any VMWare guest running Carbonite Availability should have the appropriate VMWare Tools package installed.
- **Hard links**—If you have hard links outside of the data set you are protecting, and they link to files inside the data set you are protecting, Carbonite Availability will not mirror or replicate the hard links which could lead to differences on the target.

Console requirements

The Replication Console for Linux can be run on any of the following operating systems.

- Windows 2008
- Windows 2003
- Windows 7
- Windows Vista
- Windows XP Service Pack 2 or later

Creating a files and folders job

Creating a files and folders job consists of two main tasks - creating a replication set (to identify the data to protect) and connecting that replication set to a target.

You have the following options to create a files and folders job.

- **Automated process**—If you would like to use an automated process that walks you through both the replication and connection tasks, you only need to complete the Connection Wizard steps. See *Establishing a data connection using the automated Connection Wizard* on page 84.
- **Manual process**—If you want to go through the tasks manually, begin by creating a replication set and then continue with establishing a connection. See *Creating a replication set* on page 86 and *Establishing a connection manually using the Connection Manager* on page 89.
- **NAT or firewall**—If your environment has a NAT or firewall configuration, you will need to begin with creating a replication set and then follow the instructions for establishing a NAT connection. See *Creating a replication set* on page 86 and *Establishing a connection across a NAT or firewall* on page 93.
- **Simulating a connection**—If you want to simulate a connection for planning purposes, begin by creating a replication set and then continue with establishing a simulated connection. See *Creating a replication set* on page 86 and *Simulating a connection* on page 95.

Establishing a data connection using the automated Connection Wizard

The Connection Wizard guides you through the process of protecting your data. It helps you select a source, identify the data from your source that will be included in the replication set, and select a target.

1. Start the Connection Wizard to establish your connection by selecting **Tools, Connection Wizard**.



If the Servers root is highlighted in the left pane of the Replication Console for Linux, the Connection Wizard menu option will not be available. To access the menu, expand the server tree in the left pane, and highlight a server in the tree.

2. The Connection Wizard opens to the Welcome screen. Review this screen and click **Next** to continue.



At any time while using the Connection Wizard, click **Back** to return to previous screens and review your selections.

3. If you highlighted a source in the Replication Console for Linux, the source will already be selected. If it is not, select the Carbonite Availability source. This is the server that you want to protect.



Carbonite Availability will automatically attempt to log on to the selected source using previously cached credentials. If the logon is not successful, the Logon dialog box will appear prompting for your security identification.

4. Click **Next** to continue.
5. If you highlighted a target in the Replication Console for Linux, the target will already be selected. If it is not, select the Carbonite Availability target. This is your backup server that will protect the source.



Carbonite Availability will automatically attempt to log on to the selected target using previously cached credentials. If the logon is not successful, the Logon dialog box will appear prompting for your security identification.

6. Click **Next** to continue.
7. Select **Protect data** and click **Next** to continue.
8. Choose to create a new replication set or use a replication set that already exists.
 - **Create a new replication set with this name**—If you choose to create a new replication, specify a replication set name.
 - **Use this replication set**—If you choose to use an existing replication set, specify the name of that replication set by selecting it from the pull-down menu.
9. Click **Next** to continue.

10. If you are creating a new replication set, a tree display appears identifying the volumes and directories available on your selected source server. Mark the check box of the volumes and/or directories you want to protect and click **Next** to continue.
11. Select the location on the target where the data will be stored.
 - **Send all data to a single path on the target**—This option sends all selected volumes and directories to the same location on the target. The default location is /source_name/replication_set_name/volume_name.
 - **Send all data to the same path on the target**—This option sends all selected volumes and directories to the same directories on the target.
 - **Custom**—To select a custom path, click once in the Target Path field and modify the drive and directory to the desired location.
12. Click **Next** to continue.
13. Review your selections on the summary screen. If your Connection Wizard settings are correct, establish your connection by completing one of the two options below.
 - If you do not want to set advanced options, click **Finish**. The Connection Wizard will close, the connection will be established, and mirroring and replication will begin.
 - If you want to set advanced options, click **Advanced Options**. The Connection Wizard will close and the Carbonite Availability Connection Manager will open. The **Servers** tab will be completed.

Creating a replication set

Before you can establish a connection, you must create a replication set.

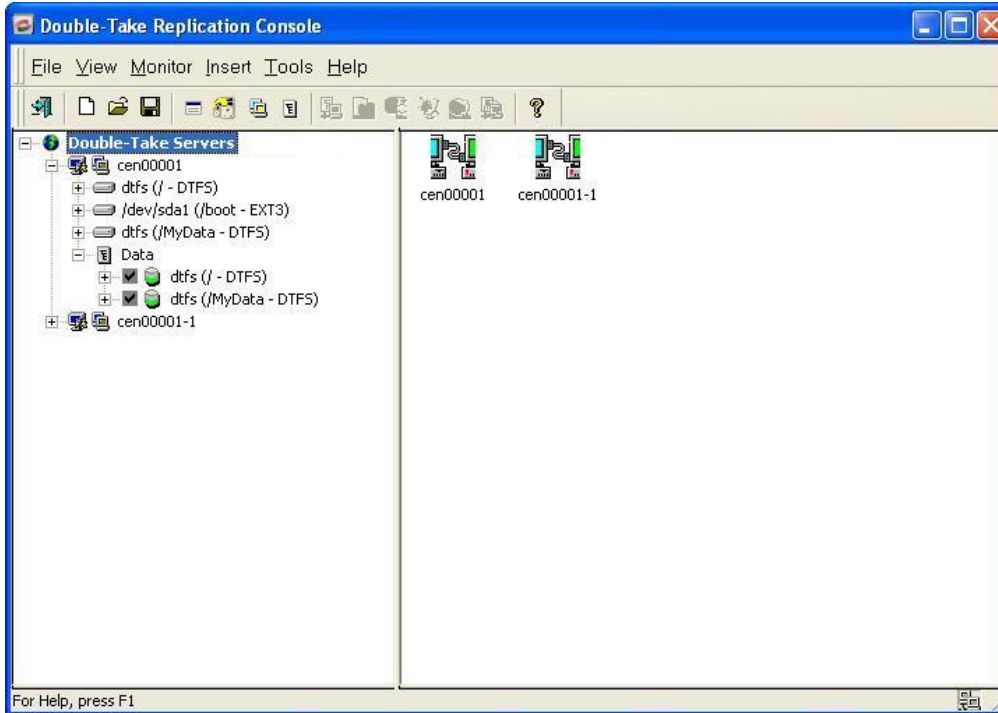
1. Highlight a source in the left pane of the Replication Console and select **Insert, Replication Set** from the menu bar. You can also right-click on the source name and select **New, Replication Set**.
2. A replication set icon appears in the left pane under the source. By default, it is named New Replication Set. Rename the newly inserted replication set with a unique name by typing over the default name and pressing **Enter**. This process is similar to naming a new folder in Windows Explorer.
3. Expand the tree under the replication set name to view the volume and directory tree for the source.



The default number of files that are listed in the right pane of the Replication Console is 2500, but this is user configurable. A larger number of file listings allows you to see more files in the Replication Console, but results in a slower display rate. A smaller number of file listings displays faster, but may not show all files contained in the directory. To change the number of files displayed, select **File, Options** and adjust the **File Listings** slider bar to the desired number.

To hide offline files, such as those generated by snapshot applications, select **File, Options** and disable **Display Offline Files**. Offline files and folders are denoted by the arrow over the lower left corner of the folder or file icon.

4. Identify the data on the source that you want to protect by selecting volumes, drives, directories, and/or specific files.



Be sure and verify what files can be included by reviewing the *Replication capabilities* on page 17.

Replication sets should only include necessary data. Including data such as temporary files, logs, and/or locks will add unnecessary overhead and network traffic. For example, if you are using Samba, make sure that the location of the lock file (lock dir in samba.conf) is not a location in your Carbonite Availability replication set.

5. After selecting the data for this replication set, right-click the new replication set icon and select **Save**. A saved replication set icon will change from red to black.
6. If you need to select a block device for replication, right-click the replication set and select **Add Device**.
7. The block devices configured for Carbonite Availability replication are shown by default. Highlight the device to include in the replication set and click **OK**.



If the device you want to include is not displayed, you can click **Show target usable devices** to view all devices which are eligible for Carbonite Availability replication. You can select any of these devices, but you cannot use them for Carbonite Availability replication until they are configured for Carbonite Availability replication.

Make sure your target has a partitioned device with sufficient space. It should be equal to or greater than the storage of the source device.

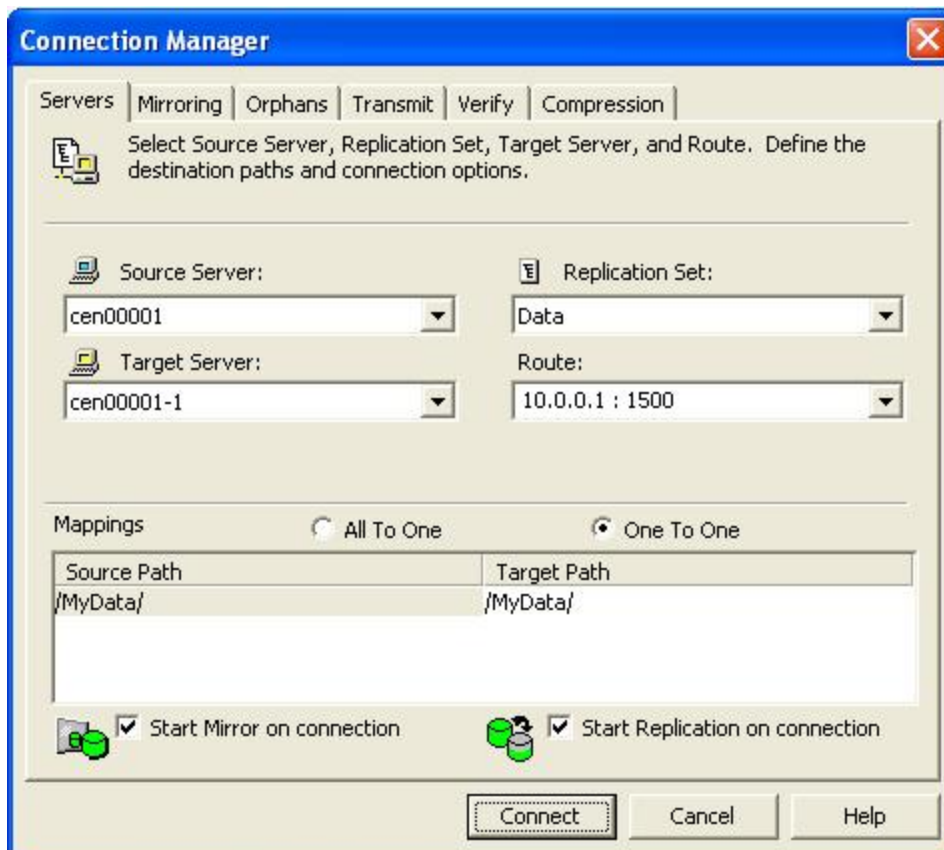
The partition size displayed may not match the output of the Linux `df` command. This is because `df` shows the size of the mounted file system not the underlying partition which may be larger. Additionally, Carbonite Availability uses powers of 1024 when computing GB, MB, and so on. The `df` command typically uses powers of 1000 and rounds up to the nearest whole value.

8. Repeat steps 6 and 7 for any additional devices.
9. Right-click the updated replication set icon and select **Save**.

Establishing a connection manually using the Connection Manager

After you have created a replication set, you can establish a connection through the Connection Manager by connecting the replication set to a target.

1. Open the Connection Manager to establish the connection.
 - Highlight the replication set and select **Tools, Connection Manager**.
 - Right-click on the replication set and select **Connection Manager**.
 - Drag and drop the replication set onto a target. The target icon could be in the left or right pane of the Replication Console for Linux.
2. The Connection Manager opens to the **Servers** tab. Depending on how you opened the Connection Manager, some entries on the **Servers** tab will be completed already. For example, if you accessed the Connection Manager by right-clicking on a replication set, the name of the replication set will be displayed in the Connection Manager. Verify or complete the fields on the **Servers** tab.



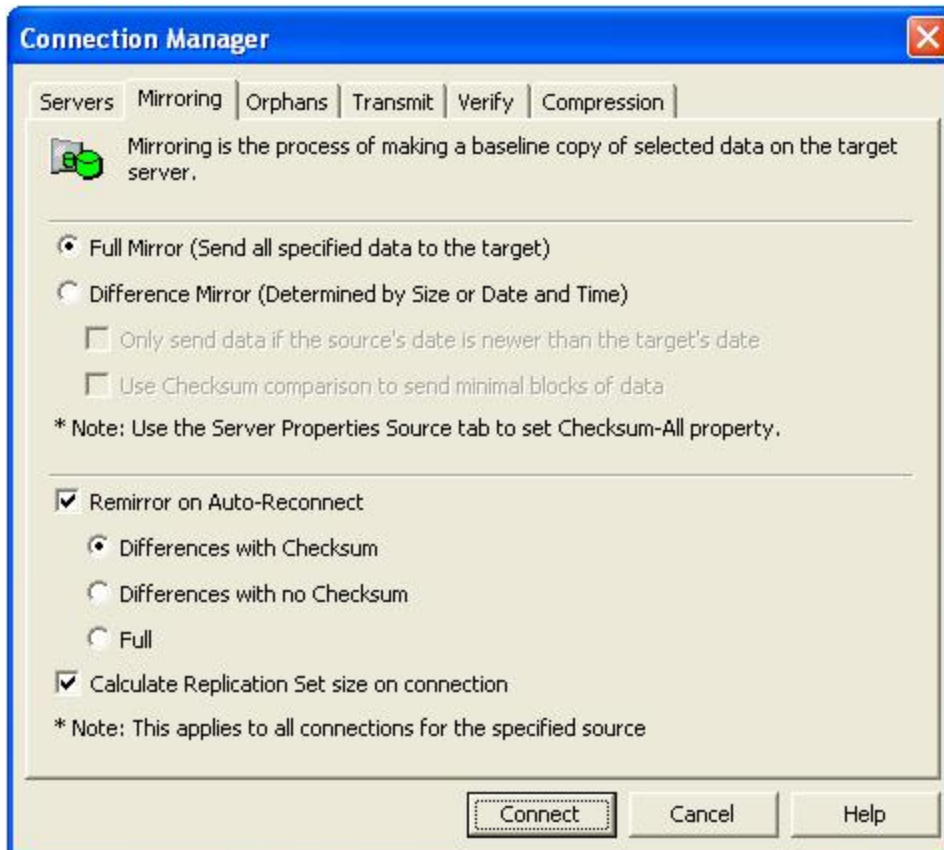
- **Source Server**—Specify the source server that contains the replication set that is going to be transmitted to the Carbonite Availability target.
- **Replication Set**—At least one replication set must exist on the source before establishing a connection. Specify the replication set that will be connected to the target.
- **Target Server**—Specify which Carbonite Availability target will maintain the copy of the source's replication set data. You can specify a machine name, IP address, or virtual IP address.
- **Route**—This is an optional setting allowing you to specify the IP address and port on the target the data will be transmitted through. This allows you to select a different route for Carbonite Availability traffic. For example, you can separate regular network traffic and Carbonite Availability traffic on a machine with multiple IP addresses.
- **Mappings**—You must specify the location on the target where the source's replication set data will reside. Carbonite Availability offers two predefined locations as well as a custom option that allows you to create your own path.
 - **All To One**—This option replicates data from the source to a single volume on the target. The pre-defined path is /source_name/replication_set_name/volume_name. If you are replicating from multiple volumes on the source, each volume would be replicated to the same volume on the target.
 - **One To One**—This option replicates data from the source to the same directory structure on the target. For example, /var/data and /usr/files on the source will be replicated to /var/data/ and /usr/files, respectively, on the target.
 - **Custom Location**—If the predefined options do not store the data in a location that is appropriate for your network operations, you can specify your own custom location where the replicated files will be sent. Click the target path and edit it, selecting the appropriate location.
- **Start Mirror on Connection**—Mirroring can be initiated immediately when the connection is established. If mirroring is not configured to start automatically, you must start it manually after the connection is established.



Data integrity cannot be guaranteed without a mirror being performed. This option is recommended for the initial connection.

- **Start Replication on Connection**—Replication can be initiated immediately when the connection is established. If replication is not configured to start automatically, you must start it manually after the connection is established. If you disable this option, you will need to perform a mirror prior to beginning replication to guarantee integrity.

- If desired, you can configure mirror settings before establishing your connection. Select the **Mirroring** tab on the Connection Manager.



- **Full Mirror**—All files in the replication set will be sent from the source to the target.
- **Difference Mirror**—Only those files that are different based size or date and time (depending on files or block devices) will be sent from the source to the target.
 - **Only send data if the source's date is newer than the target's date**—Only those files that are newer on the source are sent to the target.



If you are using a database application, do not use the newer option unless you know for certain you need it. With database applications, it is critical that all files, not just some of them that might be newer, get mirrored.

- **Use checksum comparison to send minimal blocks of data**—For those files flagged as different, the mirror performs a checksum comparison and only sends those blocks that are different.



See *Stopping, starting, pausing, or resuming mirroring* on page 115 for a comparison of how the file difference mirror settings work together, as well as how they work with the global checksum setting on the **Source** tab of the



Server Properties.

- **Remirror on Auto-Reconnect**—In certain circumstances, for example if the disk-based queues on the source are exhausted, Carbonite Availability will automatically disconnect connections (called auto-disconnect) and then automatically reconnect them (called auto-reconnect). In order to ensure data integrity on the target, Carbonite Availability will perform an automatic mirror (called an auto-remirror) after an auto-reconnect. If you enable this option, specify the type of auto-remirror that will be performed.
 - **Differences with Checksum**—Any file that is different on the source and target based on date, time, and/or size is flagged as different. The mirror then performs a checksum comparison on the flagged files and only sends those blocks that are different.
 - **Differences with no Checksum**—Any file that is different on the source and target based on date, time, and/or size is sent to the target.
 - **Full**—All files are sent to the target.



Database applications may update files without changing the date, time, or file size. Therefore, if you are using database applications, you should use the File Differences with checksum or Full option.

- **Calculate Replication Set size on connection**—Determines the size of the replication set prior to starting the mirror. The mirroring status will update the percentage complete if the replication set size is calculated.
4. Click **Connect** to establish the connection.

Establishing a connection across a NAT or firewall

If your source and target are on opposite sides of a NAT or firewall, you will need special configurations to accommodate the complex network environment. Additionally, you must have the hardware already in place and know how to configure the hardware ports. If you do not, see the reference manual for your hardware.

In this environment, you must have static mapping where a single, internal IP address is always mapped in a one-to-one correlation to a single, external IP address. Carbonite Availability cannot handle dynamic mappings where a single, internal IP address can be mapped to any one of a group of external IP addresses managed by the router.

1. Carbonite Availability uses specific ports for communication between the Carbonite Availability servers and Carbonite Availability clients. In order to use Carbonite Availability through a NAT or firewall, you must first verify the current Carbonite Availability port settings so that you can open the correct ports on your hardware to allow Carbonite Availability machines to communicate with each other. Using the following table, locate and record your port settings for each of the Carbonite Availability ports. The port setting can be found in the following locations.
 - **Replication Console for Linux**—From the Replication Console for Linux, select **File, Options**, and the **Configuration** tab.
 - **Failover for Linux console**—From the Failover for Linux console, select **Settings, Communications**.
 - **Carbonite Availability server**—From the Replication Console for Linux, right-click on a server in the tree in the left pane of the Replication Console for Linux, select **Properties**, and the **Network** tab.

Replication Console for Linux Status Transmit Port

The Status Transmit Port sends and receives directed UDP communications to display status and statistics in the Replication Console for Linux. The default setting is 1505.

Replication Console for Linux Heartbeat Advertisement

The Heartbeat Advertisement port sends and receives broadcast UDP communications to populate the Replication Console for Linux tree with Carbonite Availability servers. The default setting is 1500.

Failover for Linux console Service Transmit Port

The Service Transmit Port sends and receives TCP communication between Carbonite Availability servers and between Carbonite Availability servers and Carbonite Availability clients. The default setting is 1500.

Failover for Linux console Heartbeat Listen Port

The Heartbeat Listen Port send and receives broadcast UDP communications to populate the Failover for Linux console with Carbonite Availability servers. The default setting is 1500.

Carbonite Availability Server Service Listen Port

The Service Listen Port sends and receives TCP communication between Carbonite Availability servers and between Carbonite Availability servers and Carbonite Availability clients. The default setting is 1500.

Carbonite Availability Server Heartbeat Transmit Port

The Heartbeat Advertisement port sends and receives broadcast UDP communications to populate the Replication Console for Linux tree with Carbonite Availability servers. The default setting is 1500.

Carbonite Availability Server Status Listen Port

The Status Listen Port sends directed UDP communications to display status and statistics in the Replication Console for Linux. The default setting is 1505.

Carbonite Availability Server Statistics Logging Port

The port used for statistics logging is not available through a client. You must use the get and set DTCL commands to modify that port. See the Scripting Guide for details on the commands and the StatsPort option. The default setting is 1506.

2. You need to configure your hardware so that Carbonite Availability traffic is permitted access through the router and directed appropriately. Using the port information from the previous section, configure your router identifying each Carbonite Availability server, its IP address, and the Carbonite Availability and router ports. Also, note the following caveats.
 - Since Carbonite Availability communication occurs bidirectionally, make sure you configure your router for both incoming and outgoing traffic for all of your Carbonite Availability servers and Carbonite Availability clients.
 - In addition to UDP heartbeats, Carbonite Availability failover can use ICMP pings to determine if the source server is online. If you are going to use ICMP pings and a router between the source and target is blocking ICMP traffic, failover monitors cannot be created or used. In this situation, you must configure your router to allow ICMP pings between the source and target.

Since there are many types of hardware on the market, each can be configured differently. See your hardware reference manual for instructions on setting up your particular router.

3. If your network is configured to propagate UDP broadcasts, your servers will be populated in the Replication Console for Linux from across the router. If not, you have to manually insert the servers, by selecting **Insert, Server**. Type the IP address of the router the server is connected to and the port number the server is using for heartbeats.
4. Once your server is inserted in the Replication Console for Linux, you can use the Connection Wizard or the Connection Manager to establish your connection. See *Establishing a data connection using the automated Connection Wizard* on page 84 or *Establishing a connection manually using the Connection Manager* on page 89.

Simulating a connection

Carbonite Availability offers a simple way for you to simulate a connection in order to generate statistics that can be used to approximate the time and amount of bandwidth that the connection will use when actively established. This connection uses the TDU (Throughput Diagnostics Utility), which is a built-in null (non-existent) target to simulate a real connection. No data is actually transmitted across the network. Since there is no true connection, this connection type helps you plan for a disaster recovery solution.

1. Before and after simulating your connection, you should gather network and system information specific to Carbonite Availability operations. Use DTSetup to run DTInfo to automatically collect this data.
2. Select the DTSetup option for troubleshooting, then select the option for basic diagnostics.
3. When you run the diagnostics, it may take several minutes for it to finish processing. When it is complete, a .tar.gz file will be created in /var/cache/DT/. The file name will have DTInfo with the date and time. You must have root (or uid 0 equivalent) to execute the diagnostics or to copy or read the resulting file.
4. Opening the Connection Manager to establish the connection.
 - Highlight the replication set and select Tools, Connection Manager.
 - Right-click on the replication set and select Connection Manager.
5. The Connection Manager opens to the Servers tab. Depending on how you opened the Connection Manager, some entries on the Servers tab will be completed already. For example, if you accessed the Connection Manager by right-clicking on a replication set, the name of the replication set will be displayed in the Connection Manager. Verify or complete the fields on the Servers tab.
 - **Source Server**—Specify the source server that contains the replication set that is going to be simulated to the TDU.
 - **Replication Set**—At least one replication set must exist on the source before establishing a connection. Specify the replication set that will be connected to the TDU.
 - **Target Server**—Select the **Diagnostics** target.
 - **Route**—After selecting the **Diagnostics** target, the **Route** will automatically be populated with Throughput Diagnostics Utility (TDU).
 - **Mappings**—Mappings are not required when simulating a connection because no data is actually transmitted to the target.
 - **Start Mirror on Connection**—Make sure this option is selected so that your simulation will be realistic.
 - **Start Replication on Connection**—Make sure this option is selected so that your simulation will be realistic.
6. Click **Connect** to establish the connection. The simulation data will be logged to the Carbonite Availability statistics file.
7. Repeat steps 1-3 to run the diagnostics utility after the simulation is complete.

Protection monitoring

Most monitoring documentation is available in the *Reference Guide*, however there are a few ways to monitor protection or configure monitoring that are only available through the Replication Console for Linux.

- *Monitoring a data workload* on page 97
- *Configuring the properties of the Carbonite Availability log file* on page 106
- *Configuring the properties of the statistics file* on page 107
- *E-mailing system messages* on page 108

Monitoring a data workload

When a source is highlighted in the left pane of the Replication Console for Linux, the connections and their statistics are displayed in the right pane. Additionally, colors and icons are used for the connections, and the Carbonite Availability servers, to help you monitor your connections.

- *Connection statistics* on page 97
- *Connection and sever display* on page 102

Connection statistics

1. You can change the statistics that are displayed by selecting **File, Options** and selecting the **Statistics** tab.
2. The statistics displayed in the Replication Console for Linux will be listed with check boxes to the left of each item. Mark the check box to the left of each statistic that you want to appear, and clear the check box to the left of each statistic that you do not want to appear.
3. The statistics appear on the Replication Console for Linux in the order they appear on the **Statistics** tab. If you want to reorder the statistics, highlight the statistic to be moved and select the up or down arrow button, to the right of the vertical scroll bar, to move the selection up or down in the list. Repeat this process for each statistic that needs to be moved until you reach the desired order.
4. If you have made changes to the statistics list and have not yet saved them, you can go back to the previously used settings by clicking **Reset to Last**. This will revert the list back to the last saved settings.
5. To return the statistics list to the Carbonite Availability default selection and order, click **Reset to Default**.
6. Click **OK** to apply and save any changes that have been made to the order or display of the Replication Console for Linux statistics.

Statistics marked with an asterisk (*) are not displayed, by default.

Replication Set

Replication set indicates the name of the connected replication set.

Connection ID

The connection ID is the incremental counter used to number each connection established. This number is reset to one each time the Double-Take service is restarted.

Target Name

The name of the target as it appears in the server tree in the left pane of the Replication Console for Linux. If the server's name is not in the server tree, the IP address will be displayed.

Target IP

The target IP is the IP address on the target machine where the mirroring and replication data is being transmitted.

Target Data State

- **OK**—The data on the target is in a good state.
- **Mirroring**—The target is in the middle of a mirror process. The data will not be in a good state until the mirror is complete.
- **Mirror Required**—The data on the target is not in a good state because a remirror is required. This may be caused by an incomplete or stopped mirror or an operation may have been dropped on the target.
- **Restore required**—The data on the source and target do not match because of a failover condition. Restore the data from the target back to the source. If you want to discard the changes on the target, you can remirror to resynchronize the source and target.
- **Not Ready**—The Linux drivers have not yet completed loading on the target.

Target Status

- **OK**—The target machine is active and online.
- **Not Loaded**—The target module is not loaded on the target. (For example, the license key is invalid.)
- **Paused**—The target machine is paused by user intervention.
- **Retrying**—The target machine is retrying operations for the connection.

This field may not be updated until there is source/target activity.

Commit Mode *

The commit mode status indicates the connection status.

- **Real-time**—Data is being transmitted to the target machine in real-time.
- **Scheduled**—Data is waiting to be transmitted to the target machine until one or more transmit options have been met.

Transmit Mode

- **Started**—Data is being transferred to the target machine.
- **Paused**—If the transmission is real-time and the transmission has been paused, the **Transmit Mode** indicates **Paused**.
- **Scheduled**—If the transmission is scheduled, the **Transmit Mode** indicates **Scheduled**.
- **Stopped**—Data is not being transferred to the target machine.
- **Error**—There is a transmission error.

Mirror Status

- **Mirroring**—If the file size of the replication set has not been calculated and the data is being mirrored to the target machine, the **Mirror Status** will indicate **Mirroring**.
- **Idle**—Data is not being mirrored to the target machine.

- **Paused**—Mirroring has been paused.
- **Percentage Complete**—If the file size of the replication set has been calculated and the data is being mirrored to the target machine, the **Mirror Status** will display the percentage of the replication set that has been sent.
- **Waiting**—Mirroring is complete, but data is still being written to the target.
- **Restoring**—Data is being restored from the target to the source.
- **Verifying**—Data is being verified.
- **Removing Orphans**—Carbonite Availability is checking for orphan files within the target path location (files that exist on the target but not on the source). These files will be removed.

Replication Status

- **Replicating**—Data is being replicated to the target machine.
- **Ready**—There is no data to replicate to the target machine.
- **Stopped**—Replication has stopped.
- **Pending**—If auto-remirror is enabled and you have experienced a source or target failure and recovery, the status will change to pending while the connections are reestablished and will update when the remirror begins. If auto-remirror is disabled and you have experienced a source or target failure and recovery, replication will be Pending until a remirror is performed. Without a remirror, data integrity cannot be guaranteed.
- **Out of Memory**—Kernel memory has been exhausted.

Queued (Ops) *

The queued (ops) statistic indicates the total number of mirror and replication operations that are in the source queue.

Sent (Bytes)

The sent (bytes) statistic indicates the total number of mirror and replication bytes that have been transmitted to the target.

Sent Compressed (Bytes)

The sent compressed (bytes) statistic indicates the total number of compressed mirror and replication bytes that have been transmitted to the target. If compression is disabled, this statistic will be the same as sent (bytes).

Intermediate Queue (Bytes) *

The intermediate queue (bytes) indicates the total amount of memory being used by the operations buffer queue.

Disk Queue (Bytes)

The disk queue (bytes) indicates the amount of disk being used to queue data on the source.

Queued Replication (Bytes)

The queued replication (bytes) statistic is the total number of replication bytes that are remaining to be transmitted from the source.

Sent Replication (Bytes)

The sent replication (bytes) statistic is the total number of replication bytes that have been transmitted to the target.

Sent Compressed Replication (Bytes) *

The sent compressed replication (bytes) statistic is the total number of compressed replication bytes that have been transmitted to the target. If compression is disabled, this statistic will be the same as sent replication (bytes).

Queued Mirror (Ops) *

The queue mirror (ops) statistic is the total number of mirror operations in the queue.

Sent Mirror (Bytes)

The sent mirror (bytes) statistic is the total number of mirror bytes that have been transmitted to the target.

Sent Compressed Mirror (Bytes) *

The sent compressed mirror (bytes) statistic is the total number of compressed mirror bytes that have been transmitted to the target. If compression is disabled, this statistic will be the same as sent mirror (bytes).

Skipped Mirror (Bytes)

The skipped mirror (bytes) statistic is the total number of bytes that have been skipped when performing a difference or checksum mirror. These bytes are skipped because the data is not different on the source and target machines.

Remaining Mirror (Bytes)

The remaining mirror (bytes) statistic is the total number of mirror bytes that are remaining to be sent to the target.

Queued Replication (Ops) *

The queued replication (ops) statistic is the total number of replication operations in the queue.

Last File Touched

The last file touched identifies the last file that Carbonite Availability transmitted to the target. If you are using long file names (more than several thousand characters long) you may want to disable the display of this statistic to improve Replication Console for Linux response times.

Connected Since

Connected since is the date and time indicating when the current connection was made. This field is blank, indicating that a TCP/IP socket is not present, when the

connection is waiting on transmit options or if the transmission has been stopped. This field will maintain the date and time, indicating that a TCP/IP socket is present, when transmission has been paused.

Connection and sever display

You can configure when the icons and colors change to accommodate your network environment. For example, a slow or busy network may need longer delays before updating the icons or colors.

1. Select **File, Options**. On the **Configuration** tab, you will see **Site Monitor** and **Connection Monitor**. The **Site Monitor** fields control the icons on the left pane of the Replication Console for Linux and the icons on the right pane when a group is highlighted in the left pane. The **Connection Monitor** field controls the display when a server is highlighted in the left pane. These two separate monitoring capabilities allow for flexible monitoring.
2. Under **Site Monitor**, specify **Check Status Interval** to identify the number of seconds between requests sent from the Replication Console for Linux to the servers in order to update the display. Valid values are between 0 and 3600. The default setting is 30 seconds.
3. Under **Site Monitor**, specify **Missed Status Responses** to identify the number of responses from a server that can be missed before the Replication Console for Linux considers communications lost and updates the icons. Valid values are between 1 and 100. The default setting is 2.
4. Under **Connection Monitor**, specify **Missed Status Responses** to identify the number of responses from a server that can be missed before the Replication Console for Linux considers communications lost and updates the icons and colors. Valid values are between 0 and 1000. The default setting is 5.
5. Click **OK** to save the settings.



If the **Site Monitor** and **Connection Monitor** settings are different, at times, the icons and color may not be synchronized between the left and right panes.

The following icons are displayed in the left pane.



—An icon with yellow and blue servers indicates a server that is working properly.



—A red X on a server icon indicates the Replication Console for Linux cannot communicate with that server or that is a problem with one of the server's connections. If the connection background is gray, it is a communication issue. If the connection also has a red X, it is a connection issue.



—A red tree view (folder structure) on a server icon indicates a restore is required because of a failover.




—A black X on a server icon indicates the server is not running Carbonite Availability.

The following icons and colors are displayed in the right pane when a server is highlighted in the left pane.



—A green checkmark on a connection indicates the connection is working properly.

 —A red X on a connection indicates a connection error. For example, an error may be caused by broken transmission or pending replication. To determine the exact problem, locate the connection data item that appears in red.

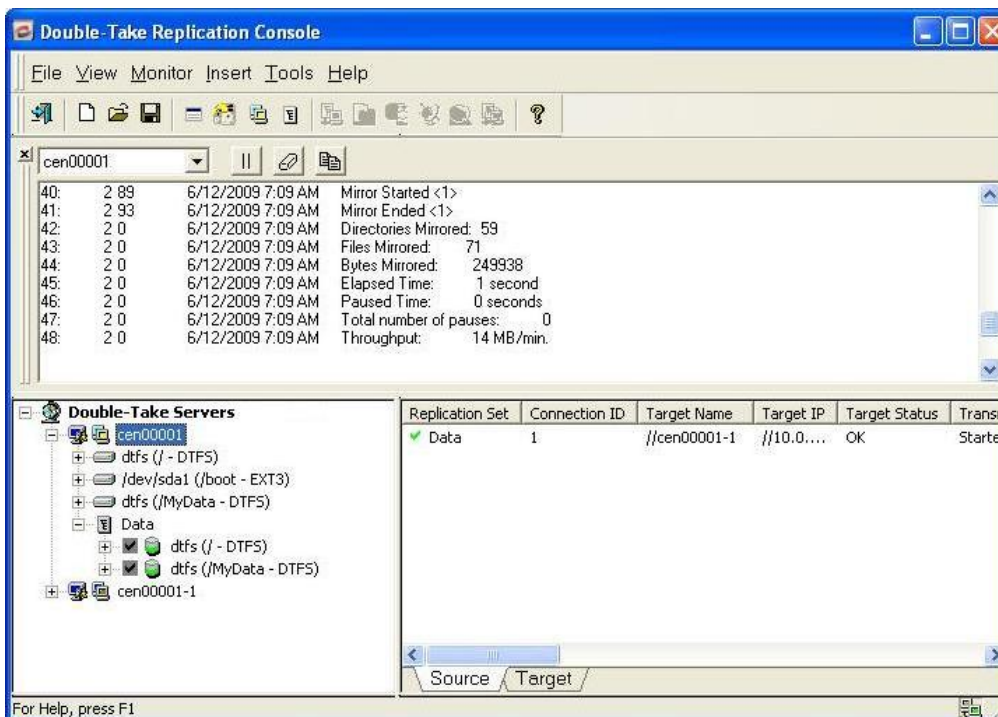
White background—If the connection background is white, the Replication Console for Linux and the source are communicating.

Gray background—If the connection background is gray, the Replication Console for Linux and the source are no longer communicating. The connection data stops updating once communications have stopped. Once communications have been reestablished, the connection background will change back to white.

Viewing the Carbonite Availability log file through the Replication Console for Linux

In addition to the statistics and status shown in the Replication Console for Linux, you can also open a message window to view the Carbonite Availability log file.

1. Open a new message window using any of the following methods.
 - Right-click on the server that you want to monitor in the left pane and select **New, Message Window**.
 - Select the Message Window icon from the toolbar.
 - Select **Monitor, New Message Window** and identify the **Server** that you want to monitor.
2. Repeat step 1 if you want to open multiple message windows.



The standard appearance of the message window is a white background. If your message window has a gray background, the window is inactive. The Replication Console for Linux may have lost communications with that server, for example, or you may no longer be logged into that server.

The message window is limited to the most recent 1000 lines. If any data is missing an entry in red will indicate the missing data. Regardless of the state of the message window, all data is maintained in the Carbonite Availability log on the server.

3. To control the window after it is created, use one of the following methods to access the control methods listed in the following table.
 - Right-click on the message window and select the appropriate control.
 - Select the appropriate toolbar control.

- Select **Monitor**, the name of the message window, and the appropriate control.

Close 

Closes the message window

Clear 

Clears the message window

Pause/Resume 

Pauses and resumes the message window.

Pausing prevents new messages from being displayed in the message window so that you are not returned to the bottom of the message window every time a new message arrives. The messages that occur while the window is logged are still logged to the Carbonite Availability log file.

Resuming displays the messages that were held while the window was paused and continues to display any new messages.

Pausing is automatically initiated if you scroll up in the message window. The display of new log messages will automatically resume when you scroll back to the bottom.

Copy 

Allows you to copy selected text

Options

This control is only available from the **Monitor** menu. Currently, there are no filter options available so this option only allows you to select a different server. In the future, this control will allow you to filter which messages to display.

4. To change which server you are viewing messages for, select a different machine from the drop down list on the toolbar. If necessary, the login process will be initiated.
5. To move the message window to other locations on your desktop, click and drag it to another area or double-click it to automatically undock it from the Replication Console for Linux.

Configuring the properties of the Carbonite Availability log file

1. To modify the maximum file size and the number of Carbonite Availability log files that are maintained, access the Server Properties dialog box by right-clicking a machine name in the left pane of the Replication Console for Linux and selecting **Properties**.
2. Select the **Logging** tab.
3. At the top of the window, **Folder** indicates the directory where the log files are located. The default is the directory where the Carbonite Availability program files are installed.
4. Modify any of the options under **Messages and Alerts**, if necessary.
 - **Maximum Length**—Specify the maximum length of the log file. The default size is 1048576 bytes and is limited by the available hard drive space.
 - **Maximum Files**—Specify the maximum number of log files that are maintained. The default is 5 and the maximum is 999.

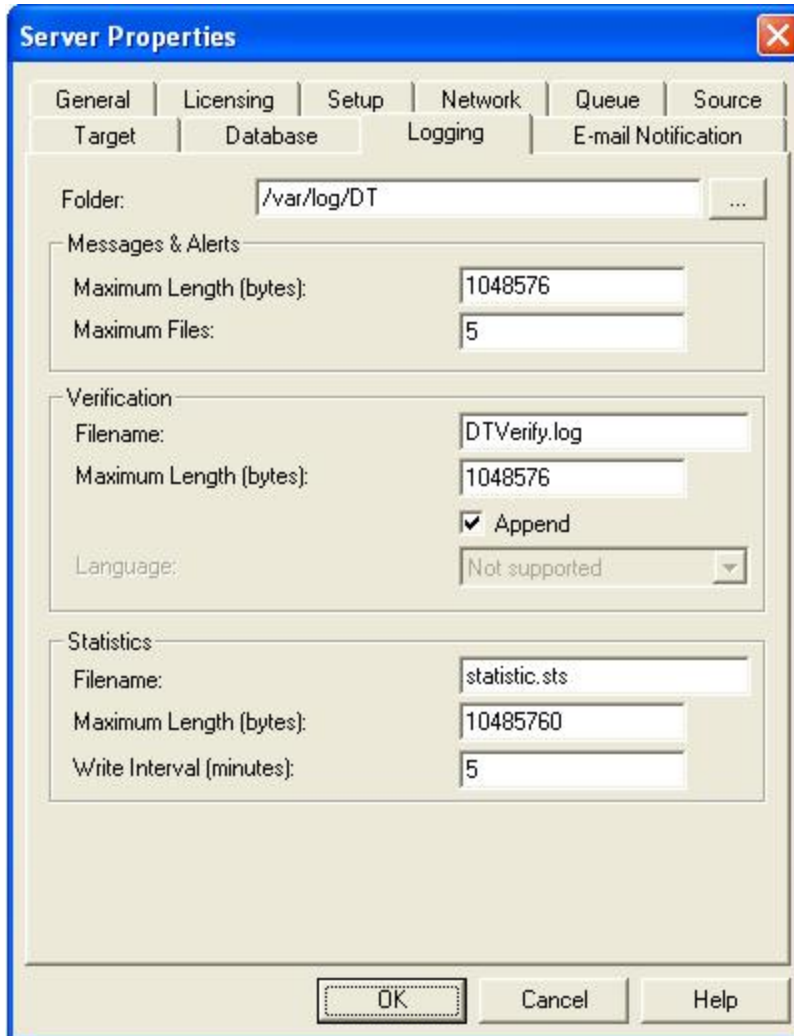


If you change the **Maximum Length** or **Maximum Files**, you must restart the Double-Take service for the change to take effect.

5. Click **OK** to save the changes.

Configuring the properties of the statistics file

1. Right-click a machine in the left pane of the Replication Console for Linux and select **Properties**.
2. Select the **Logging** tab.

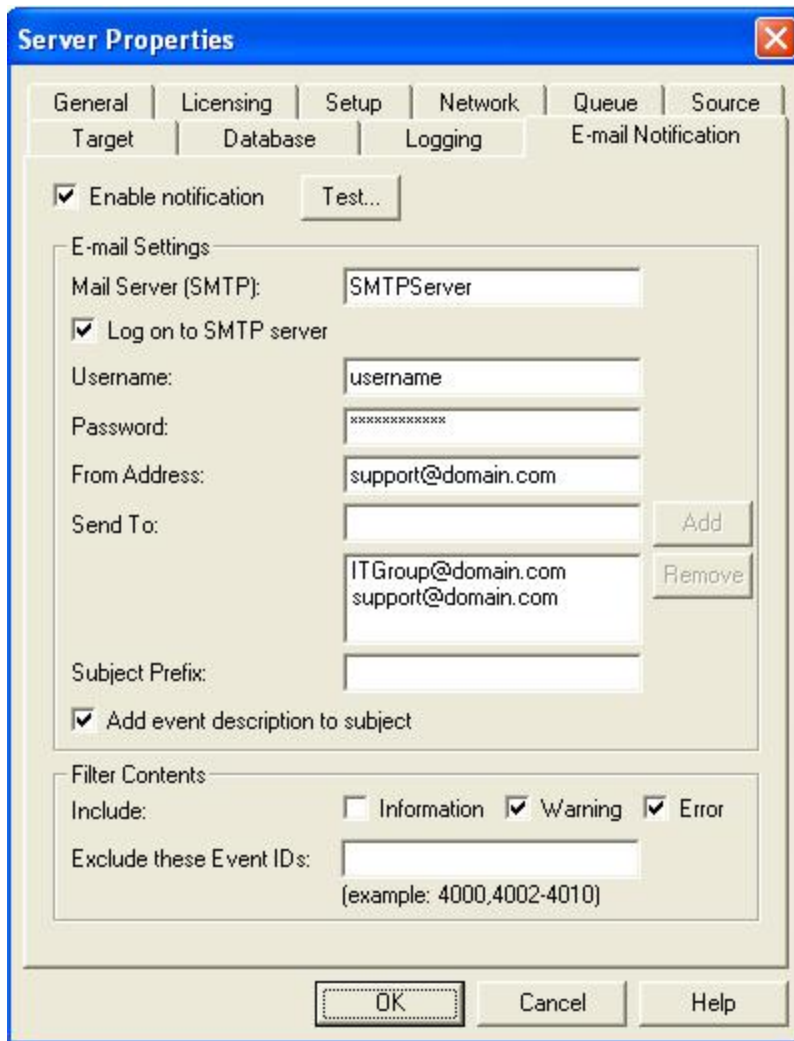


3. At the top of the tab, specify the **Folder** where the log files for messages, alerts, verification, and statistics will be saved.
4. Under **Statistics**, specify the following information.
 - **Filename**—The name of the statistics log file. The default file name is statistic.sts.
 - **Maximum Length**—The maximum length of the statistics log file. The default maximum length is 10 MB. Once this maximum has been reached, Carbonite Availability begins overwriting the oldest data in the file.
 - **Write Interval**—The frequency in which Carbonite Availability writes the statistical data to the statistics log file. The default is every 5 minutes.
5. Select the **Setup** tab.
6. Verify that **Log Statistics Automatically** is enabled. If disabled, statistics will not be logged.
7. Click **OK** to save the settings.

E-mailing system messages

You can e-mail system messages to specified addresses. The subject of the e-mail will contain an optional prefix, the server name where the message was logged, the message ID, and the severity level (information, warning, or error). The text of the message will be displayed in the body of the e-mail message.

1. To enable e-mail notification for a server, right-click the server in the left pane of the Replication Console and select **Properties**.
2. Select the **E-mail Notification** tab.



The screenshot shows the 'Server Properties' dialog box with the 'E-mail Notification' tab selected. The 'Enable notification' checkbox is checked, and a 'Test...' button is visible. Under 'E-mail Settings', the 'Mail Server (SMTP)' is set to 'SMTPServer', and the 'Log on to SMTP server' checkbox is checked. The 'Username' is 'username' and the 'Password' is masked with 'xxxxxxxxxx'. The 'From Address' is 'support@domain.com'. The 'Send To' list contains 'ITGroup@domain.com' and 'support@domain.com', with 'Add' and 'Remove' buttons. The 'Subject Prefix' is empty, and the 'Add event description to subject' checkbox is checked. Under 'Filter Contents', the 'Include' section has 'Warning' and 'Error' checked, and 'Information' unchecked. The 'Exclude these Event IDs' field is empty, with an example '(example: 4000,4002-4010)' below it. The 'OK', 'Cancel', and 'Help' buttons are at the bottom.

3. Select **Enable notification**.



Any specified notification settings are retained when **Enable notification** is disabled.

4. Specify your e-mail settings.

- **Mail Server (SMTP)**—Specify the name of your SMTP mail server.
-



Specifying an SMTP server is the preferred method because it provides a direct connection between the mail server and Carbonite Availability, which decreases message latency and allows for better logging when the mail server cannot be reached.

If you do not specify an SMTP server, Carbonite Availability will attempt to use the Linux mail command. The success will depend on how the local mail system is configured. Carbonite Availability will be able to reach any address that the mail command can reach.

- **Log on to SMTP Server**—If your SMTP server requires authentication, enable **Log on to SMTP Server** and specify the **Username** and **Password** to be used for authentication. Your SMTP server must support the LOGIN authentication method to use this feature. If your server supports a different authentication method or does not support authentication, you may need to add the Carbonite Availability server as an authorized host for relaying e-mail messages. This option is not necessary if you are sending exclusively to e-mail addresses that the SMTP server is responsible for.
- **From Address**—Specify the e-mail address that you want to appear in the From field of each Carbonite Availability e-mail message. The address is limited to 256 characters.
- **Send To**—Specify the e-mail address that each Carbonite Availability e-mail message should be sent to and click **Add**. The e-mail address will be inserted into the list of addresses. Each address is limited to 256 characters. You can add up to 256 e-mail addresses. If you want to remove an address from the list, highlight the address and click **Remove**. You can also select multiple addresses to remove by Ctrl-clicking.
- **Subject Prefix** and **Add event description to subject**—The subject of each e-mail notification will be in the format Subject Prefix : Server Name : Message Severity : Message ID : Message Description. The first and last components (Subject Prefix and Message Description) are optional. The subject line is limited to 150 characters.

If desired, enter unique text for the **Subject Prefix** which will be inserted at the front of the subject line for each Carbonite Availability e-mail message. This will help distinguish Carbonite Availability messages from other messages. This field is optional.

If desired, enable **Add event description** to subject to have the description of the message appended to the end of the subject line. This field is optional.

- **Filter Contents**—Specify which messages that you want to be sent via e-mail. Specify **Information**, **Warning**, and/or **Error**. You can also specify which messages to exclude based on the message ID. Enter the message IDs as a comma or semicolon separated list. You can indicate ranges within the list.
-



You can test e-mail notification by specifying the options on the E-mail Notification tab and clicking **Test**. If desired, you can send the test message to a different e-mail address by selecting **Send To** and entering a comma or semicolon separated list of addresses. Modify the message text up to 1024 characters, if necessary. Click **Send** to test the e-mail notification. The results will be displayed in a message box.

Click **OK** to close the message and click **Close** to return to the E-mail Notification tab. 

If an error occurs while sending an e-mail, a message will be generated. This message will not trigger an e-mail. Subsequent e-mail errors will not generate additional messages. When an e-mail is sent successfully, a message will then be generated. If another e-mail fails, one message will again be generated. This is a cyclical process where one message will be generated for each group of failed e-mail messages, one for each group of successful e-mail messages, one for the next group of failed messages, and so on.

If you start and then immediately stop the Double-Take service, you may not get e-mail notifications for the log entries that occur during startup.

By default, most virus scan software blocks unknown processes from sending traffic on port 25. You need to modify the blocking rule so that Carbonite Availability e-mail messages are not blocked.

Connections

A unique connection ID is associated with each Carbonite Availability connection. The connection ID provides a reference point for each connection. The connection ID is determined by sequential numbers starting at one (1). Each time a connection is established, the ID counter is incremented. It is reset back to one each time the Double-Take service is restarted. For example, if the Double-Take service was started and the same replication set was connected to five target machines, each connection would have a unique connection ID from 1 to 5. The connection can be in various states.

- **Started**—The network connection exists and is available for data transmission. Replication and mirror data are transmitted to the target as soon as possible. This is the standard state that you will see most often.
- **Stopped**—Carbonite Availability has linked the source and target, but the network connection does not exist. Replication and mirror data are not transmitted to the target but are held in queue on the source.
- **Paused**—The network connection exists and is available for data transmission, but the replication and mirror data is being held in a queue and is not being transmitted to the target.
- **Scheduled**—Carbonite Availability has linked the source and target, but the network connection is not established until event driven or scheduling criteria have been met.
- **Error**—A transmission error has occurred. Possible errors include a broken physical line or a failed target service.

You can perform the following functions to manage your connections.

- *Pausing and resuming target processing* on page 112
- *Disconnecting a connection* on page 113

Pausing and resuming target processing

You can break the source/target connection without disconnecting the connection, so that you can control the transmission of data across the network. You can do this by pausing the target. If the target is paused, data is queued on the source until you manually resume the target. For example, you may want to pause the target while you perform a backup of the target data, and then resume the target when the backup is complete.

While the target is paused, the Carbonite Availability source cannot queue data indefinitely. If the source queue is filled, Carbonite Availability will automatically disconnect the connections and attempt to reconnect them.

To pause a target, right-click a target server on the left pane of the Replication Console for Linux and select **Pause Target**. All active connections to that target will complete the operations already in progress. You will see **Pause Pending** in the Replication Console for Linux while these operations are completed. The status will update to **Paused** after the operations are completed. Any new operations will be queued on the source until the target is resumed. When you are ready to resume the target, right-click the target and select **Resume Target**.



If you have multiple connections to the same target, all connections will be paused and resumed.

Disconnecting a connection

To disconnect a Carbonite Availability connection, right-click the connection on the right pane of the Replication Console for Linux and select **Disconnect**. The source and target will be disconnected.



If a connection is disconnected while large amounts of data still remain in queue, the Replication Console for Linux may become unresponsive while the data is being flushed. The Replication Console for Linux will respond when all of the data has been flushed from the queue.

If a connection is disconnected and the target is monitoring the source for failover, you will be prompted if you would like to continue monitoring for a failure. If you select **Yes**, the Carbonite Availability connection will be disconnected, but the target will continue monitoring the source. To make modifications to the failure monitoring, you will need to use the Failover for Linux console. If you select **No**, the Carbonite Availability connection will be disconnected, and the source will no longer be monitored for failure by the target.

Mirroring

Mirroring is one of the key components of Carbonite Availability. You can perform the following functions to manage mirroring.

- *Stopping, starting, pausing, or resuming mirroring* on page 115
- *Mirroring automatically* on page 117
- *Removing orphan files* on page 119

Stopping, starting, pausing, or resuming mirroring

After a connection is established, you need to be able to control the mirroring. You can start, stop, pause and resume mirroring. Right-click the connection on the right pane of the Replication Console for Linux and select **Mirroring** and the appropriate mirror control.

- **Pause or Resume**—When pausing a mirror, Carbonite Availability stops queuing mirror data on the source but maintains a pointer to determine what information still needs to be mirrored to the target. Therefore, when resuming a paused mirror, the process continues where it left off.
- **Stop**—When stopping a mirror, Carbonite Availability stops queuing mirror data on the source and does not maintain a pointer to determine what information still needs to be mirrored to the target. Therefore, when starting a mirror that has been stopped, the process will mirror all of the data contained in the replication set.
- **Start**—If you select to start a mirror, you will need to make the following two selections on the Start Mirror dialog box.
 - **Full Mirror**—All files in the replication set will be sent from the source to the target.
 - **File differences**—Only those files that are different based size or date and time will be sent from the source to the target. Expand *File difference mirror options compared* below to see how the file difference mirror settings work together, as well as how they work with the global checksum setting on the **Source** tab of the Server Properties.
 - **Send data only if Source is newer than Target**—Only those files that are newer on the source are sent to the target.



If you are using a database application, do not use the newer option unless you know for certain you need it. With database applications, it is critical that all files, not just some of them that might be newer, get mirrored.

- **Use block checksum**—For those files flagged as different, the mirror performs a checksum comparison and only sends those blocks that are different.
- **Calculate Replication Set size prior to mirror**—Determines the size of the replication set prior to starting the mirror. The mirroring status will update the percentage complete if the replication set size is calculated.

File difference mirror options compared

- **File Differences**—Any file that is different on the source and target based on the date, time, and/or size is transmitted to the target. The mirror sends the entire file.
- **File Differences and Only if Source is Newer**—Any file that is newer on the source than on the target based on date and/or time is transmitted to the target. The mirror sends the entire file.
- **File Differences and Checksum**—This option is dependent on the global checksum all option on the Server Properties source tab.
 - **Checksum All disabled**— Any file that is different on the source and target based on date, time, and/or size is flagged as different. The mirror then performs a checksum comparison on the flagged files and only sends those blocks that are different.
 - **Checksum All enabled**—The mirror performs a checksum comparison on all files and only sends those blocks that are different.
- **File Differences, Only if Source is Newer, and Checksum**—Any file that is newer on the source than on the target based on date and/or time is flagged as different. The mirror then performs a checksum comparison on the flagged files and only sends those blocks that are different.

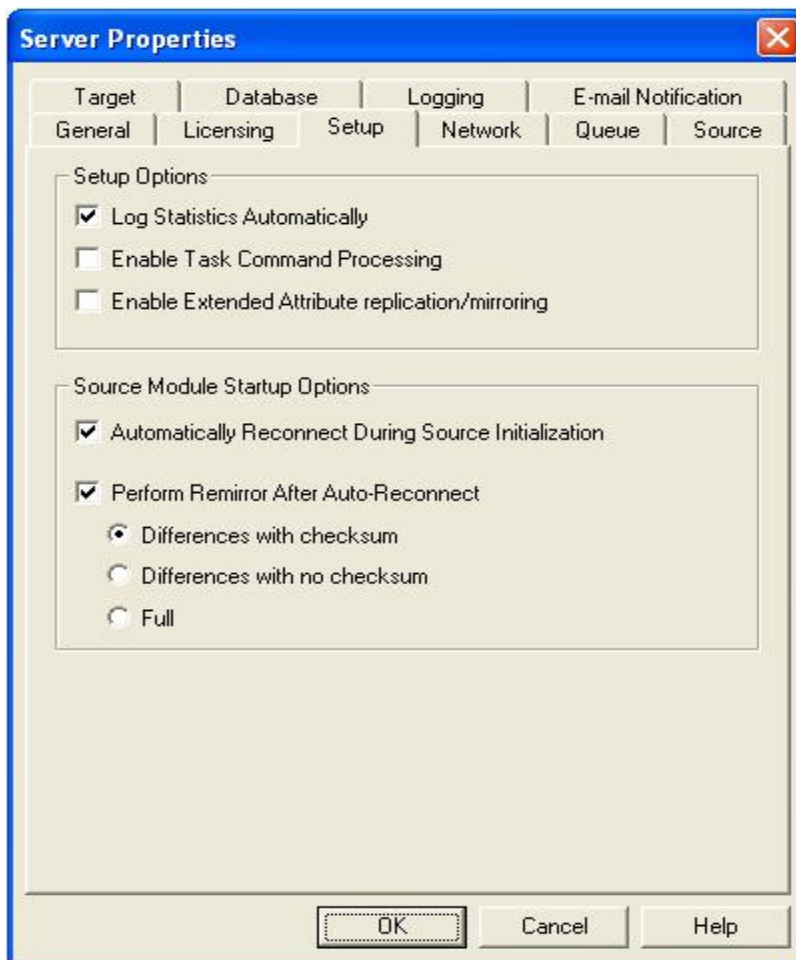
Mirroring automatically

In certain circumstances, for example if the disk-based queues on the source are exhausted, Carbonite Availability will automatically disconnect connections (called auto-disconnect) and then automatically reconnect them (called auto-reconnect). In order to ensure data integrity on the target, Carbonite Availability will perform an automatic mirror (called an auto-remirror) after an auto-reconnect.



Auto-remirror is a per source option. When enabled, all connections from the source will perform an auto-remirror after an auto-reconnect. When disabled, none of the connections from the source will perform an auto-remirror after an auto-reconnect.

1. Right-click a server in the left pane of the Replication Console for Linux and select **Properties**.
2. Select the **Setup** tab.



3. Verify that the **Perform Remirror After Auto-Reconnect** check box is selected to initiate an auto-remirror after an auto-reconnect.



If auto-remirror is disabled and an auto-reconnect occurs, the transmission state of the connection will remain pending after the reconnect until a mirror is started manually.

4. Specify the type of mirror that you wish to perform.

- **Differences with Checksum**—Any file that is different on the source and target based on date, time, and/or size is flagged as different. The mirror then performs a checksum comparison on the flagged files and only sends those blocks that are different.
 - **Differences with no Checksum**—Any file that is different on the source and target based on date, time, and/or size is sent to the target.
 - **Full**—All files are sent to the target.
-



Database applications may update files without changing the date, time, or file size. Therefore, if you are using database applications, you should use the Differences with checksum or Full option.

Stopping, starting, pausing, or resuming mirroring contains a comparison of how the file difference remirror settings work together, as well as how they work with the global checksum setting on the **Source** tab of the Server Properties.

5. Click **OK** to save the settings.

Removing orphan files

An orphan file is a file that exists in the target's copy of the replication set data, but it does not exist in the source replication set data. An orphan file can be created when you delete a file contained in the source replication set while there is no Carbonite Availability connection. For example, if a connection was made and a mirror was completed and then the connection was stopped and a file was deleted on the source, an orphan file will exist on the target. Because the connection has been disconnected, the delete operation is not replicated to the target and the file is not deleted on the target. Additionally, orphan files may also exist if files were manually copied into or deleted from the location of the target's copy of the replication set data.

You can configure orphan files to be moved or deleted automatically during a mirror, verify, or restore, or you can move or delete orphan files manually at any time. You can move or delete all orphan files on the target or only those orphan files that are older than a specified period of time. The results of orphan processing are maintained in the Carbonite Availability log on the target, including the number of moved/deleted orphan files, the directories, and the number of bytes.

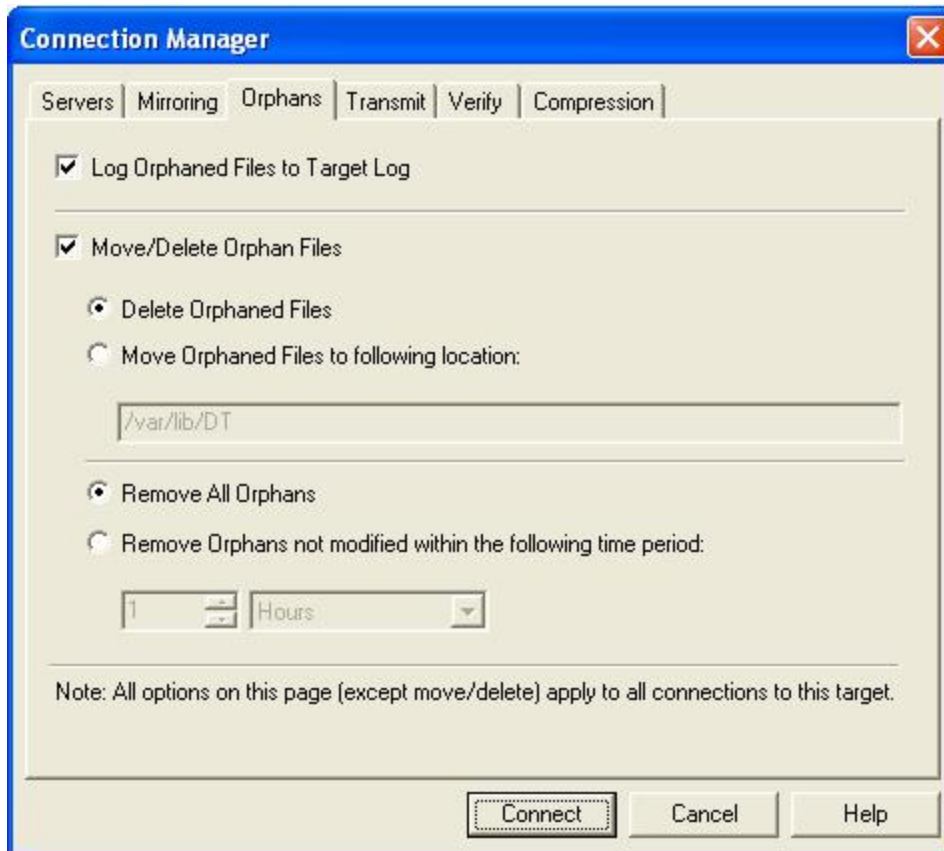


Orphan file configuration is a per target option. All connections to the same target will have the same orphan file configuration.

If Carbonite Availability is configured to move orphan files, the Carbonite Availability log file will indicate that orphan files have been deleted even though they have actually been moved. This is a reporting issue only.

If delete orphans is enabled, carefully review any replication set rules that use wildcard definitions. If you have specified wildcards to be excluded from your replication set, files matching those wildcards will also be excluded from orphan file processing and will not be deleted from the target. However, if you have specified wildcards to be included in your replication, those files that fall outside the wildcard inclusion rule will be considered orphans and will be deleted from the target.

-
1. If you want to preview which files are identified as orphan files, right-click an established connection and select **Remove Orphans, Preview**. Check the log file on the target for the list of orphaned files.
 2. If you want to remove orphan files manually, right-click an established connection and select **Remove Orphans, Start**.
 3. If you want to stop the process after it has been started, right-click the connection and select **Remove Orphans, Stop**.
 4. To configure orphan files for processing during a mirror, verify, or restore, use the following instructions.
 - a. Right-click the connection on the right pane of the Replication Console for Linux and select **Connection Manager**.
 - b. Select the **Orphans** tab.



- c. Specify if you want to log the name of the orphan files to the Carbonite Availability log file on the target by marking **Log Orphaned Files to Target Log**.
- d. By default, the orphan files feature is disabled. To enable it, mark **Move/Delete Orphan Files**.
- e. Specify if you want to **Delete Orphaned Files** or **Move Orphaned Files** to a different location. If you select the move option, identify the location where these orphan files will be located.



If you are moving or deleting orphan files, select a move location outside of the replication set. If you select the location where the files are currently located, the files will be deleted. If you select another location inside the replication set, the files will be moved multiple times and then possibly deleted.

- f. Specify if you want to **Remove All Orphans** or **Remove Orphaned Files not modified within the following time period**. If you select the time-based option, only orphans older than the time you specify will be removed.
- g. Click **OK** to save the settings.

Replication

Replication is one of the key components of Carbonite Availability. This section contains the following replication topics.

- *Replication capabilities* on page 17—Review this list to learn what Carbonite Availability supports for replication.
- *Replication sets* on page 122—This section contains instructions for creating and using Carbonite Availability replication sets.
- *Starting replication* on page 136—Since replication is one of the key components of Carbonite Availability, this topic includes instructions for starting replication.
- *Inserting tasks during replication* on page 137—You can insert tasks to be processed inline with replication.

Replication sets

A replication set defines the data on a source machine that Carbonite Availability protects. Replication sets are defined by volumes, directories, files, or wild card combinations. Creating multiple replication sets allows you to customize sets of data that need to be protected.

When a replication set is created, a series of rules are defined that identify the volumes, directories, files, and/or wild card combinations that will be replicated to the target. Each rule includes:

- **Path**—The path including volume, drive, directory, file, and/or wild card
- **Include**—If the specified path is to be included in the files sent to the target
- **Exclude**—If the specified path is not to be included in the files sent to the target
- **Recursive**—If the rule should automatically be applied to the subdirectories of the specified path

For example, a replication set rule might be `volume\directory* inc, rec`

This specifies that all files contained in the `volume\directory` path are included in the replication set. Because recursion is set, all files and subdirectories under `volume\directory` are also included. A complete replication set becomes a list of replication set rules.

Replication sets offer flexibility tailoring Carbonite Availability to your environment. For example, multiple replication sets can be created and saved for a source to define a unique network configuration. There may be three replication sets - Critical Data, User Data, and Offsite Data. Critical Data could be configured to replicate, in real-time, to an onsite high-availability server. Offsite Data is replicated across a WAN and, therefore, is configured to queue changes until a sufficient amount of data is changed to justify transmission. At that point, the connection is made and stays active until all the data is transmitted. User Data is not replicated throughout the day, but a nightly changed file mirror copies only blocks of data that are different between the source and target server prior to a nightly tape backup operation being run on the target server. Each of these replication sets can be automated to transmit as needed, thus protecting your entire environment.

Keep in mind the following notes when creating and working with replication sets and connections.

- **Limitations**

- Replication set rules are limited in length meaning that the entire `volume\directory\filename` including slashes, spaces, periods, extensions, cannot exceed 259 characters.
- Carbonite Availability can mirror, replicate, verify, and restore paths up to 4094 characters. Paths longer than 4094 characters will be skipped and logged to the Carbonite Availability log file and the Linux system log.
- Do not name replication sets or select a target location using illegal characters. Illegal characters include the following.
 - period .
 - question mark ?
 - forward or backward angle bracket < >
 - colon :
 - quotation mark "
 - forward or backward slash \ /
 - asterisk *
 - pipe or vertical bar |

- **Error checking and avoidance**

- Do not connect more than one replication set to the same location on a target. You could overwrite or corrupt your data.
- Replication sets contain error checking to avoid inadvertent overwrites of the replication set rules. When replication sets are modified, a generation number is associated with the modifications. The generation number is incremented anytime the modifications are saved, but the save is not allowed if there is a mismatch between the generation number on the source and the Replication Console for Linux. You will be notified that the replication set could not be saved. This error checking safeguards the replication set data in the event that more than one client machine is accessing the source's replication sets.
- Carbonite Availability will not replicate the same data from two different replication sets on your source. The data will only be replicated from one of the replication sets. If you need to replicate the same data more than once, connect the same replication set to multiple targets.
- If you rename the root folder of a connected replication set, Carbonite Availability interprets this operation as a move from inside the replication set to outside the replication set. Therefore, since all of the files under that directory have been moved outside the replication set and are no longer a part of the replication set, those files will be deleted from the target copy of the replication set. This, in essence, will delete all of your replicated data from the target. If you have to rename the root directory of your replication set, make sure that the replication set is not connected.
- When creating replication sets, keep in mind that when recursive rules have the same type (include or exclude) and have the same root path, the top level recursive rule will take precedence over lower level non-recursive rules. For example, if you have `/var/data` included recursively and `/var/data/old` included nonrecursively, the top level rule, `/var/data/`, will take precedence and the rule `/var/data/old` will be discarded. If the rules are different types (for example, `/var/data` is included and `/var/data/old` is excluded), both rules will be applied as specified.

- **Virus protection**

- Virus protection software on the target should not scan replicated data. If the data is protected on the source, operations that clean, delete, or quarantine infected files will be replicated to the target by Carbonite Availability. If the replicated data on the target must be scanned for viruses, configure the virus protection software on both the source and target to delete or quarantine infected files to a different directory that is not in the replication set. If the virus software denies access to the file because it is infected, Carbonite Availability will continually attempt to commit operations to that file until it is successful, and will not commit any other data until it can write to that file.

Creating a replication set

Before you can establish a connection, you must create a replication set.

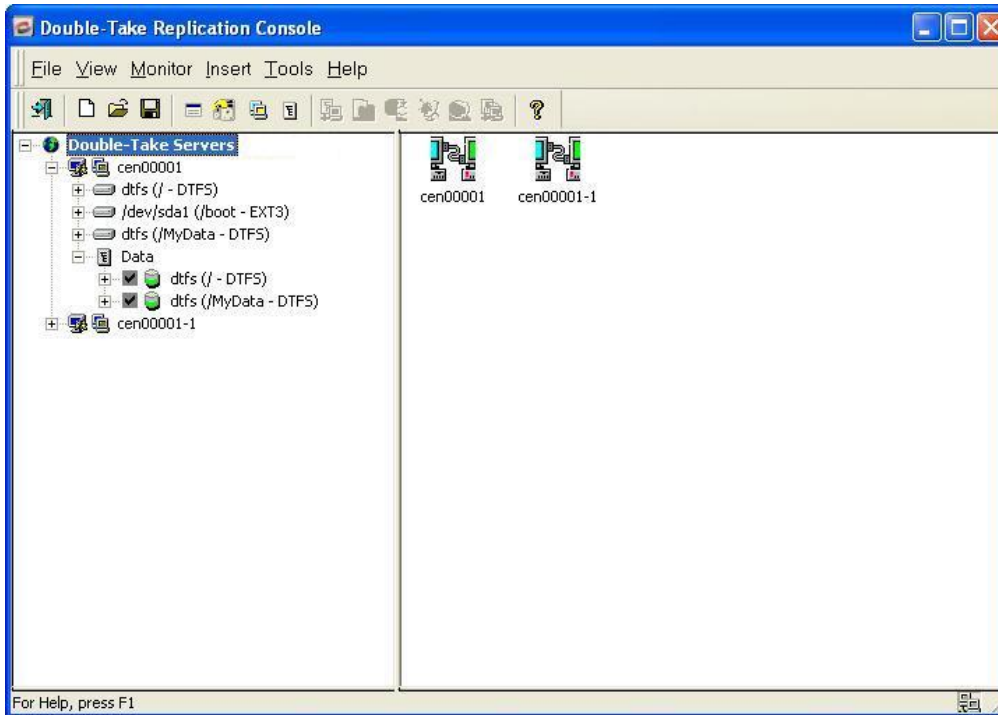
1. Highlight a source in the left pane of the Replication Console and select **Insert, Replication Set** from the menu bar. You can also right-click on the source name and select **New, Replication Set**.
2. A replication set icon appears in the left pane under the source. By default, it is named New Replication Set. Rename the newly inserted replication set with a unique name by typing over the default name and pressing **Enter**. This process is similar to naming a new folder in Windows Explorer.
3. Expand the tree under the replication set name to view the volume and directory tree for the source.



The default number of files that are listed in the right pane of the Replication Console is 2500, but this is user configurable. A larger number of file listings allows you to see more files in the Replication Console, but results in a slower display rate. A smaller number of file listings displays faster, but may not show all files contained in the directory. To change the number of files displayed, select **File, Options** and adjust the **File Listings** slider bar to the desired number.

To hide offline files, such as those generated by snapshot applications, select **File, Options** and disable **Display Offline Files**. Offline files and folders are denoted by the arrow over the lower left corner of the folder or file icon.

4. Identify the data on the source that you want to protect by selecting volumes, drives, directories, and/or specific files.



Be sure and verify what files can be included by reviewing the *Replication capabilities* on page 17.

Replication sets should only include necessary data. Including data such as temporary files, logs, and/or locks will add unnecessary overhead and network traffic. For example, if you are using Samba, make sure that the location of the lock file (lock dir in samba.conf) is not a location in your Carbonite Availability replication set.

5. After selecting the data for this replication set, right-click the new replication set icon and select **Save**. A saved replication set icon will change from red to black.
6. If you need to select a block device for replication, right-click the replication set and select **Add Device**.
7. The block devices configured for Carbonite Availability replication are shown by default. Highlight the device to include in the replication set and click **OK**.



If the device you want to include is not displayed, you can click **Show target usable devices** to view all devices which are eligible for Carbonite Availability replication. You can select any of these devices, but you cannot use them for Carbonite Availability replication until they are configured for Carbonite Availability replication.

Make sure your target has a partitioned device with sufficient space. It should be equal to or greater than the storage of the source device.

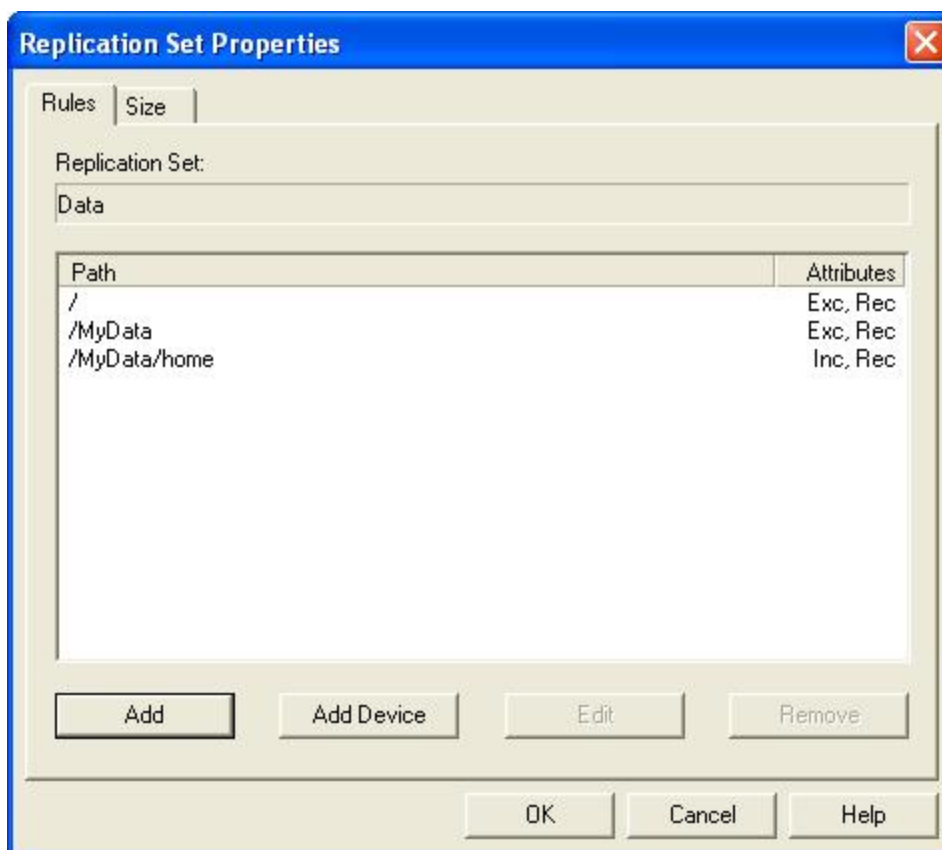
The partition size displayed may not match the output of the Linux `df` command. This is because `df` shows the size of the mounted file system not the underlying partition which may be larger. Additionally, Carbonite Availability uses powers of 1024 when computing GB, MB, and so on. The `df` command typically uses powers of 1000 and rounds up to the nearest whole value.

8. Repeat steps 6 and 7 for any additional devices.
9. Right-click the updated replication set icon and select **Save**.

Creating or modifying replication rules manually

There may be times when you cannot browse for data when creating a replication set. For example, you can create a replication set rule for a directory or file that does not exist. Since you cannot browse for the location, you have to create replication set rule manually. At other times, the data you want to replicate cannot be easily selected from the Replication Console for Linux. For example, you may want to select all .db files from a specific volume or directory. This task may be easier to complete by creating the replication set rule manually. Use the following instructions to create or modify a replication set rule manually.

1. If you do not have a replication set created, you need to create one. Highlight a source in the left pane of the Replication Console for Linux and select **Insert, Replication Set** from the menu bar. You can also right-click on the source name and select **New, Replication Set**. A replication set icon appears in the left pane under the source. By default, it is named New Replication Set. Rename the newly inserted replication set with a unique name by typing over the default name and pressing **Enter**. This process is similar to naming a new folder in Windows Explorer.
2. Right-click on the replication set icon and select **Properties**. The Replication Set Properties dialog box appears and lists any existing rules. The existing rules may have been entered manually or selected by browsing the source. Each rule will display the attributes associated it.



- **Inc**—Include indicates that the specified path is to be included in the files sent to the target
- **Exc**—Exclude indicates that the specified path is not to be included in the files sent to the target

- **Rec**—Recursion indicates the rule should automatically be applied to the subdirectories of the specified path. If you do not select this option, the rule will not be applied to subdirectories.
3. From the Replication Set Properties dialog box, click **Add**.
 4. Specify a path, wild card, or specific file name. Select the **Include**, **Exclude**, and/or **Recurse sub-directories** attributes to be applied to this rule and click **OK**.
 5. If you need to select block devices for replication, click **Add Device**. The block devices configured for Carbonite Availability replication are shown by default. Highlight the device to include in the replication set and click **OK**. If the device you want to include is not displayed, you can click **Show target usable devices** to view all devices which are eligible for Carbonite Availability replication. You can select any of these devices, but you cannot use them for Carbonite Availability replication until they are configured for Carbonite Availability replication.
 6. If you need to edit an existing rule, highlight it and click **Edit**.
 7. If you need to remove a rule, highlight it and click **Remove**.
 8. After the replication set rules have been defined, exit the Replication Set Properties dialog box by clicking **OK**. Notice the replication set icon has changed from black to red, indicating changes to the replication set rules. If you click **Cancel**, your changes will not be reflected in the current replication set.
 9. Right-click the replication set icon and select **Save**. A saved replication set icon will change from red to black.

Selecting a block device for replication

Carbonite Availability allows you to select block devices for replication.

1. In the left pane, right-click the replication set that should include the block device and select **Add Device**.
2. The block devices configured for Carbonite Availability replication are shown by default. Highlight the device to include in the replication set and click **OK**.



If the device you want to include is not displayed, you can click **Show target usable devices** to view all devices which are eligible for Carbonite Availability replication. You can select any of these devices, but you cannot use them for Carbonite Availability replication until they are configured for Carbonite Availability replication.

Make sure your target has a partitioned device with sufficient space. It should be equal to or greater than the storage of the source device.

The partition size displayed may not match the output of the Linux `df` command. This is because `df` shows the size of the mounted file system not the underlying partition which may be larger. Additionally, Carbonite Availability uses powers of 1024 when computing GB, MB, and so on. The `df` command typically uses powers of 1000 and rounds up to the nearest whole value.

3. Repeat steps 1 and 2 for any additional devices.

Modifying a replication set

Carbonite Availability allows you to make modifications to a replication set when you want to change the data you wish to protect. This allows you to add, remove, or modify any replication set rules without having to create a new replication set.

1. In the left pane, highlight the replication set you want to modify and expand the volume and directory levels as needed.
2. Modify the items by marking or clearing the volume, drive, directory, or file check boxes. Notice the replication set icon has changed from black to red, indicating changes to the replication set rules.
3. After updating the rules for this replication set, right-click the replication set icon and select **Save**. A saved replication set icon will change from red to black.



If you save changes to a connected replication set, it is recommended that you perform a mirror to guarantee data integrity between the source and target machines. A dialog box will appear instructing you to disconnect and reconnect the replication set and perform a difference mirror.

Renaming and copying a replication set

To rename or copy a replication set, click once on a highlighted replication set name to edit the field. Specify a unique name and press **Enter**. The process is similar to renaming a folder in Windows Explorer. If the original replication set has not been saved (red icon), the new name replaces the original name. If the original replication set is saved (black icon), the new name creates a copy of the original replication set.

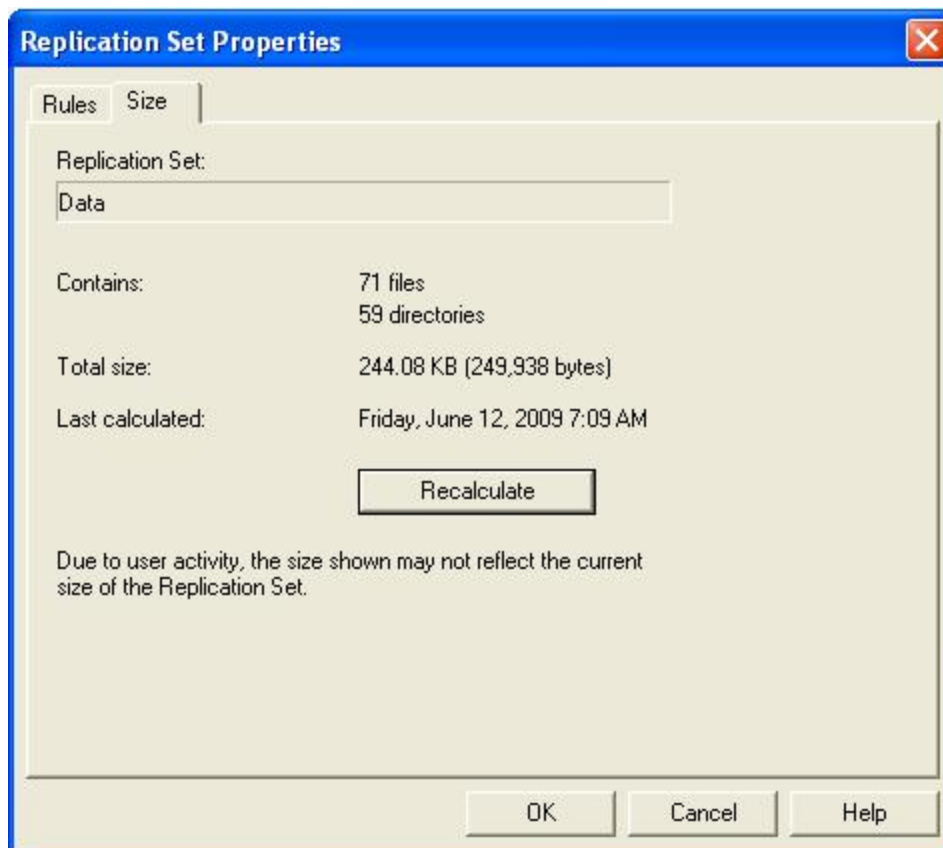


If you save changes to a connected replication set, it is recommended that you perform a mirror to guarantee data integrity between the source and target machines. A dialog box will appear instructing you to disconnect and reconnect the replication set and perform a difference mirror.

Calculating replication set size

While Carbonite Availability is mirroring, the right pane of the Replication Console for Linux displays statistics to keep you informed of its progress. If the size of the replication set is determined before the mirror is started, Carbonite Availability can display the percentage of the replication set that has been mirrored in the **Mirror Status** column. If the size was not calculated prior to starting the mirror, the column displays **Mirroring**.

1. Right-click on the replication set icon and select **Properties**. The Replication Set Properties dialog box appears.
2. Select the **Size** tab.



3. If the replication set size has never been determined, click **Calculate**. If the replication set has previously been determined, the button will be labeled **Recalculate**. Depending on user activity, the size shown may not accurately reflect the current size of the replication set. If changes are occurring to files in the replication set while the calculation is being made, the actual size may differ slightly. The amount of data is determined at the exact time the calculation is made.
4. Click **OK** to return to the Replication Console for Linux.



You can also configure the replication set calculation when establishing a connection through the Connection Manager by selecting Calculate Replication Set size on connection on the Mirroring tab.

If your replication set contains a large number of files, for example, ten thousand or more, you may want to disable the calculation of the replication set size so that data will start being mirrored sooner. If calculation is enabled, the source calculates the file size before it starts mirroring. This can take a significant amount of time depending on the number of files and system performance. Disabling calculation will result in the mirror status not showing the percentage complete or the number of bytes remaining to be mirrored.

Exporting and importing a replication set

To help reuse replication sets between servers, you can export an existing replication set on one server and import it on another.

- **Exporting a replication set**—Right-click an existing replication set and select **Export**. Select a location and file name for the replication set information, and click **Save**. If you want to share the replication set information with other consoles, select a location accessible by other consoles.
- **Importing a replication set**—Right-click the server where you want to import the replication set and select **New, Import Replication Set**. Locate the replication set information file and click **Open**. By default, the new replication set will have the same name as the original replication set. If desired, modify the name. Press Enter to accept the replication set name. By default, the new replication set is imported in an unsaved state. An unsaved replication set icon is red. Modify the replication set definition (include or exclude volumes or files) and then save the replication set by right-clicking on it and selecting **Save**. A saved replication set icon is black.

Deleting a replication set

You can only delete a replication set if it is not currently connected. If the replication set is connected, you must disconnect the connection and then delete the replication set.

To delete a replication set, right-click the replication set icon and select **Delete**. Additionally, you can highlight the replication set and press the **Delete** key on the keyboard.

Starting replication

Starting replication when establishing a connection is the default and recommended configuration. If replication is not started, data is not added to the queue on the source, and source/target data integrity is not guaranteed.

To start replication, right-click the connection on the right pane of the Replication Console for Linux and select **Replication, Start**. After starting replication, you should perform a remirror to guarantee the source and target data are identical.

Inserting tasks during replication

Task command processing is a Carbonite Availability feature that allows you to insert and run tasks at various points during the replication of data. Because the tasks are user-defined, you can achieve a wide variety of goals with this feature. For example, you might insert a task to create a snapshot or run a backup on the target after a certain segment of data from the source has been applied on the target. This allows you to coordinate a point-in-time backup with real-time replication.

Task command processing can be enabled from the Replication Console for Linux, but it can only be initiated through the scripting language. See the *Scripting Guide* for more information.

To enable task command processing from the Replication Console for Linux, right-click a server in the left pane of the Replication Console for Linux, select **Properties**, select the **Setup** tab, and select **Enable Task Command Processing**.



If you disable this option on a source server, you can still submit tasks to be processed on a target, although task command processing must be enabled on the target.

Verification

Verification is the process of confirming that the data on the target is identical to the data on the source. Verification creates a log file detailing what was verified as well as which files are not synchronized. If the data is not the same, Carbonite Availability can automatically initiate a remirror. The remirror ensures data integrity between the source and target.

- *Verifying manually* on page 139—You can verify your data at any time manually.
- *Verifying on a schedule* on page 140—You can schedule verification tasks for periodic intervals.
- *Configuring the verification log* on page 142—You can configure how the verification information is logged.



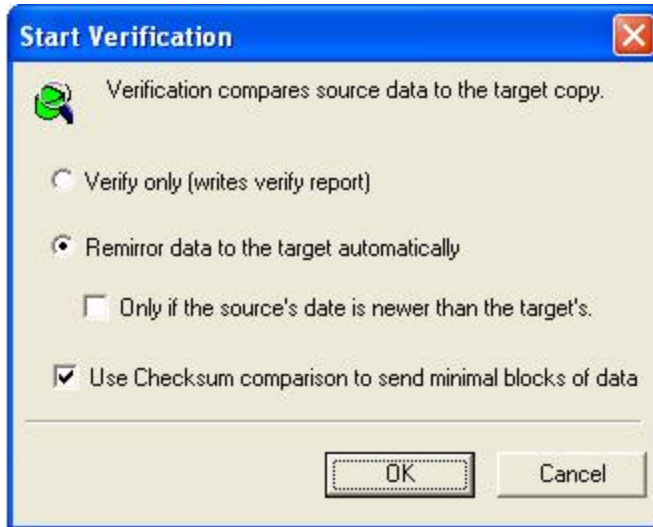
Differences in files on the source and target should be expected for files and applications that are in use during the verification process.

The verification report will not display the full attributes for hard links.

Verifying manually

A manual verification can be run anytime a mirror is not in progress.

1. Right-click the connection on the right pane of the Replication Console for Linux and select **Verify**.
2. Select the verification options that you would like to perform.



- **Verify only**—This option verifies the data and generates a verification log, but it does not remirror any files that are different on the source and target.
- **Remirror data to the target automatically**—This option verifies the data, generates a verification log, and remirrors to the target any files that are different on the source.
- **Only if the source's date is newer than the target's**—If you are remirroring your files, you can specify that only files that are newer on the source than the target be remirrored.



If you are using a database application, do not use the newer option unless you know for certain you need it. With database applications, it is critical that all files, not just some of them that might be newer, get mirrored.

- **Use Checksum comparison to send minimal blocks of data**—Specify if you want the verification process to use a block checksum comparison to determine which blocks are different. If this option is enabled, only those blocks (not the entire files) that are different will be identified in the log and/or remirrored to the target.



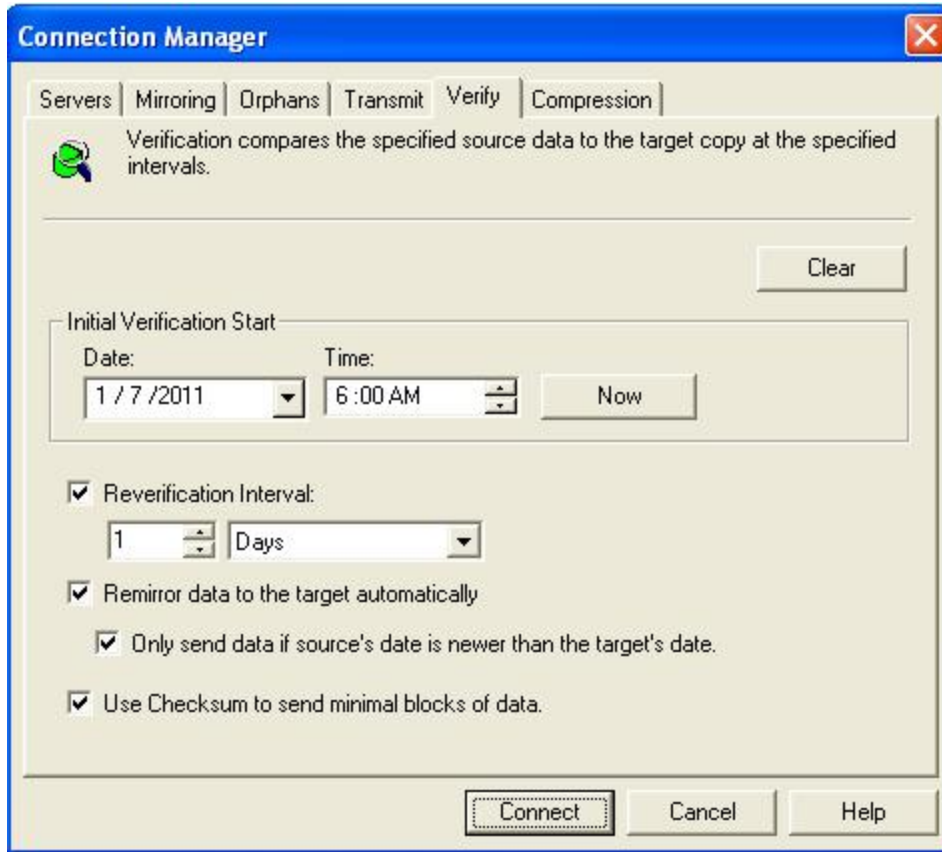
Database applications may update files without changing the date, time, or file size. Therefore, if you are using database applications, you should use the block checksum comparison to ensure proper verification and remirroring.

3. Click **OK** to start the verification.

Verifying on a schedule

Verification can be scheduled to occur automatically at periodic intervals.

1. Right-click the connection on the right pane of the Replication Console for Linux and select **Connection Manager**.
2. Select the **Verify** tab.



3. Specify when you want to start the initial verification. Select the immediate date and time by clicking **Now**, or enter a specific **Date** and **Time**. The down arrow next to **Date** displays a calendar allowing easy selection of any date. **Time** is formatted for any AM or PM time.
4. Mark the **Reverification Interval** check box to repeat the verification process at the specified interval. Specify an amount of time and choose minutes, hours, or days.
5. Select if you want to **Remirror data to the target automatically**. When enabled, Carbonite Availability will verify the data, generate a verification log, and remirror to the target any files that are different on the source. If disabled, Carbonite Availability will verify the data and generate a verification log, but no files will be remirrored to the target.
6. If you are remirroring your files, you can specify **Only send data if source's date is newer than the target's date** so that only files that are newer on the source than on the target are remirrored.



If you are using a database application, do not use the newer option unless you know for certain you need it. With database applications, it is critical that all files, not just some of them that might be newer, get mirrored.

7. Specify if you want the verification process to **Use Checksum to send minimal blocks of data** to determine which blocks are different. If this option is enabled, only those blocks (not the entire files) that are different will be identified in the log and/or remirrored to the target.
-



Database applications may update files without changing the date, time, or file size. Therefore, if you are using database applications, you should use the block checksum comparison to ensure proper verification and remirroring.

8. Click **OK** to save the settings.
-

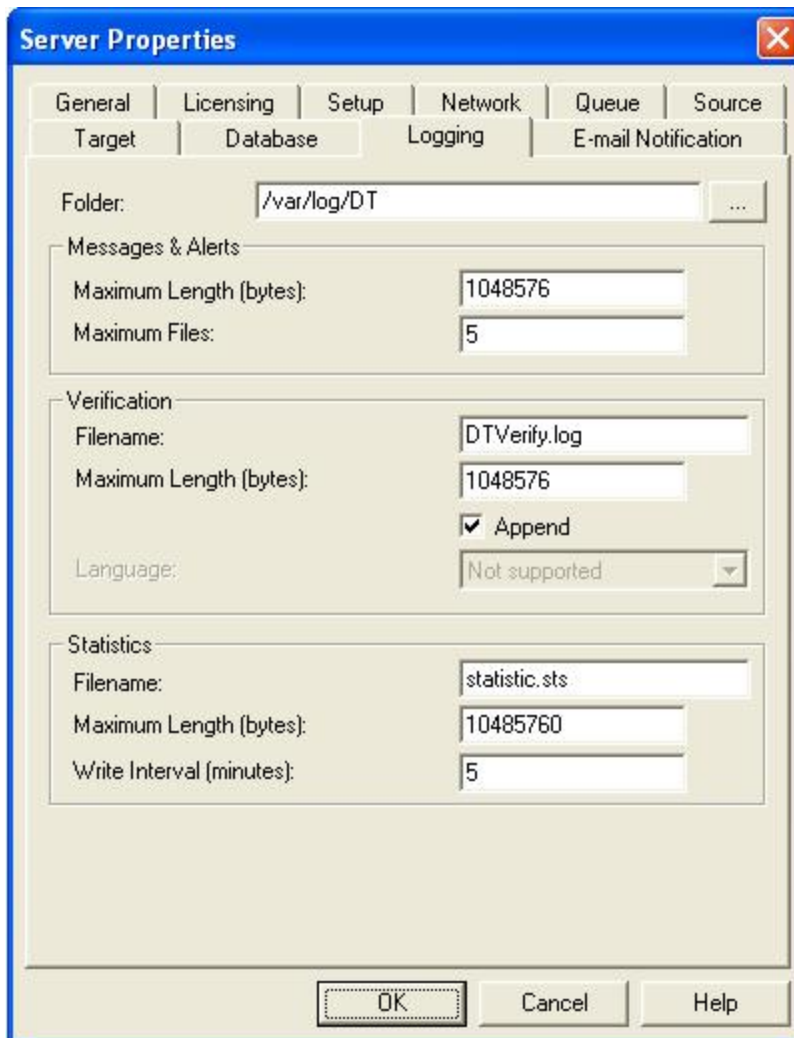


When you schedule a verification process, it may run a verification report when you save the scheduled verification settings. The scheduled verification will still process as expected.

Configuring the verification log

A verification log is created on the source during the verification process. The log identifies what is verified as well as which files are not synchronized.

1. Right-click the source server on the left pane of the Replication Console for Linux and select **Properties**.
2. Select the **Logging** tab.



3. At the top of the window, **Folder** identifies the location where the log files identified on this tab are stored. By default, the log files are stored in the same directory as the Carbonite Availability program files.
4. Under the Verification section, **Filename** contains the base log file name for the verification process. The replication set name will be prepended to the base log file name. For example, since the default is DTVerify.log, the verification log for the replication set called UserData would be UserData DTVerify.log.
5. Specify the **Maximum Length** of the log file. The default is 1048576 bytes (1 MB). When the log file reaches this limit, no additional data will be logged.

6. By default, the log is appended to itself each time a verification process is completed. Clear the **Append** check box if you do not want to append to the previous log file.
-



Changes made to the verification log in the **Server Properties, Logging** tab will apply to all connections from the current source machine.

7. Specify the **Language** of the log file. Currently, English is the only available language.
8. Click **OK** to save the settings.

In the log file, each verification process is delineated by beginning and end markers. A list of files that are different on the source and target is provided as well cumulative totals for the verification process. The information provided for each file is the state of its synchronization between the source and the target at the time the file is verified. If the remirror option is selected so that files that are different are remirrored, the data in the verify log reflects the state of the file before it is remirrored, and does not report the state of the file after it is remirrored. If a file is reported as different, review the output for the file to determine what is different.

Data transmission

Carbonite Availability data is continuously transmitted to the target machine. Although the data may be queued if the network or target machine is slow, the default transmission setting is to transmit the data as soon as possible. You can modify the transmission to suit your environment.

- *Stopping, starting, pausing, and resuming transmission* on page 145—You can maintain the source/target connection, but still control the transmission of data across the network by using the manual transmission controls. If transmission is paused, the data is queued on the source until you manually restart the transmission.
- *Scheduling data transmission* on page 145—You can set event driven or scheduling criteria to determine when data is transmitted. Data is queued on the source until the event or schedule is met. Also, transmission can be stopped by using these criteria. Scheduled transmission options can be toggled on and off, allowing you to enable them only when you need to use them.
- *Limiting transmission bandwidth* on page 150—You can specify bandwidth limitations to restrict the amount of network bandwidth used for Carbonite Availability data transmissions. Data is queued on the source until bandwidth is available. Bandwidth limitations can be full-time or scheduled.
- *Compressing data for transmission* on page 152—You can compress data to reduce the amount of bandwidth needed to transmit Carbonite Availability data.

Stopping, starting, pausing, and resuming transmission

To start, pause, or resume the transmission of data from the source to the target, right-click an established connection and select **Transmit** and the appropriate transmission control.

Scheduling data transmission

Using the Connection Manager **Transmit** tab, you can set start and stop criteria along with a schedule window.



Carbonite Availability checks the schedule once every second, and if a user-defined criteria is met, transmission will start or stop, depending on the option specified.

Any replication sets from a source connected to the same IP address on a target will share the same scheduled transmission configuration.

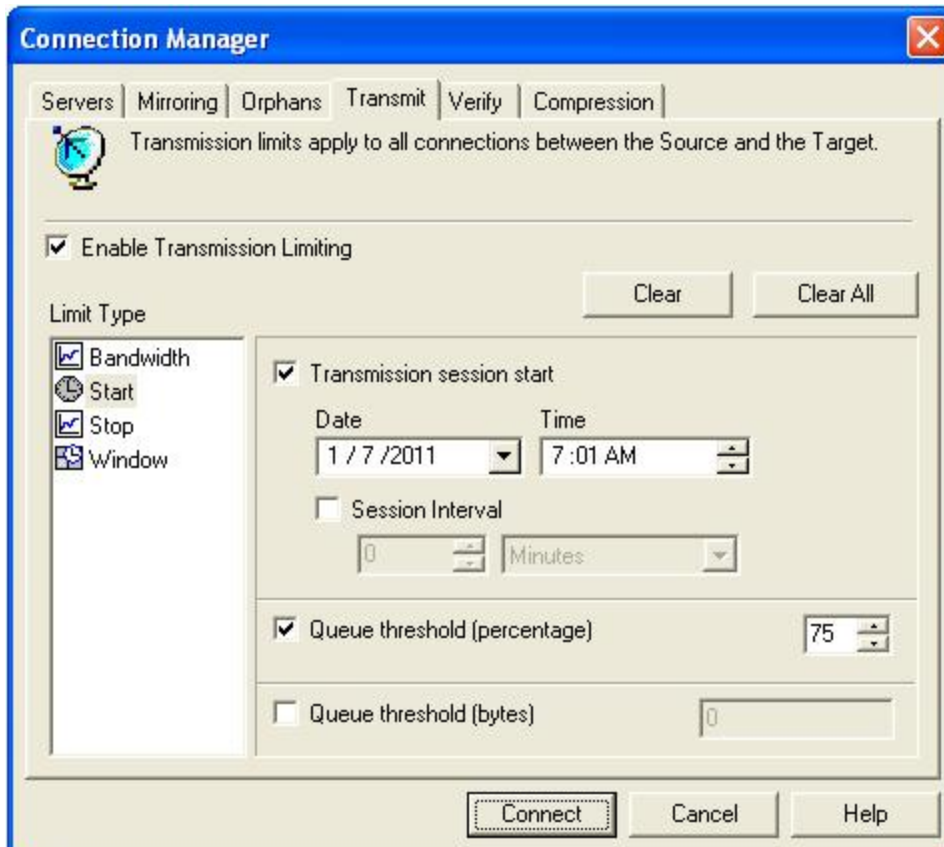
1. Right-click the connection on the right pane of the Replication Console for Linux and select **Connection Manager**.
2. Select the **Transmit** tab. The **Transmit** tab contains four limit types: **Bandwidth**, **Start**, **Stop**, and **Window**. The transmission options for each limit type are displayed by highlighting a selection in the **Limit Type** box.

At the top of the **Transmit** tab dialog box, the **Enable Transmission Limiting** check box allows you to turn the transmission options on or off. You can enable the transmission options by marking the **Enable Transmission Limiting** check box when you want the options to be applied, but you can disable the transmission options, without losing the settings, by clearing that check box.

Also at the top of the **Transmit** tab dialog box, the **Clear All** button, when selected, will remove all transmission limitations that have been set under any of the limit types. The **Clear** button will clear the settings only for the **Limit Type** selected.

3. When you schedule transmission start criteria, transmission will start when the criteria is met and will continue until the queue is empty or a transmission stop criteria is met. Select the **Start option** in the Limit Type box.

Define the start options for Carbonite Availability transmission by using any combination of the following options.



- **Transmission session start**—This option establishes a date and time of the day to begin transmitting data. For example, you may want to specify a transmission time that corresponds to a low bandwidth usage time. Once started, Carbonite Availability will continue to transmit data until the queue is empty or until another limitation stops the transmission. Specify a **Date** and **Time** to start transmitting data. The down arrow next to the date field displays a calendar allowing easy selection of any date. The time field is formatted for any AM or PM time.
- **Session Interval**—This option begins transmitting Carbonite Availability data at specified intervals of time. This option is used in conjunction with **Transmission session start**. For example, if the **Session Interval** is set to repeat transmission every 30 minutes and the **Transmission session start** is set to begin transmitting at 10 p.m., if the queue is emptied at 10:20 the transmission will stop. The start criteria is again met at 10:30 and Carbonite Availability will begin transmitting any new data in the queue. Specify an interval for additional transmissions by indicating a length of time and choosing minutes, hours, or days.
- **Queue Threshold (percentage)**—If the allocated amount of queue disk space is in use, Carbonite Availability cannot continue to queue data causing an auto-disconnect and the potential for loss of data. To avoid using the entire queue, you can configure Carbonite Availability to begin transmitting data to the target when the queue reaches a certain percentage. For example, if you specify 40%, when 40% of the queue is in use, Carbonite Availability initiates the transmission process and sends the data in the queue to the target machine. The transmission stops when the queue is empty or a Carbonite Availability stop transmission criteria is met. Specify a percentage of the disk queue and system memory

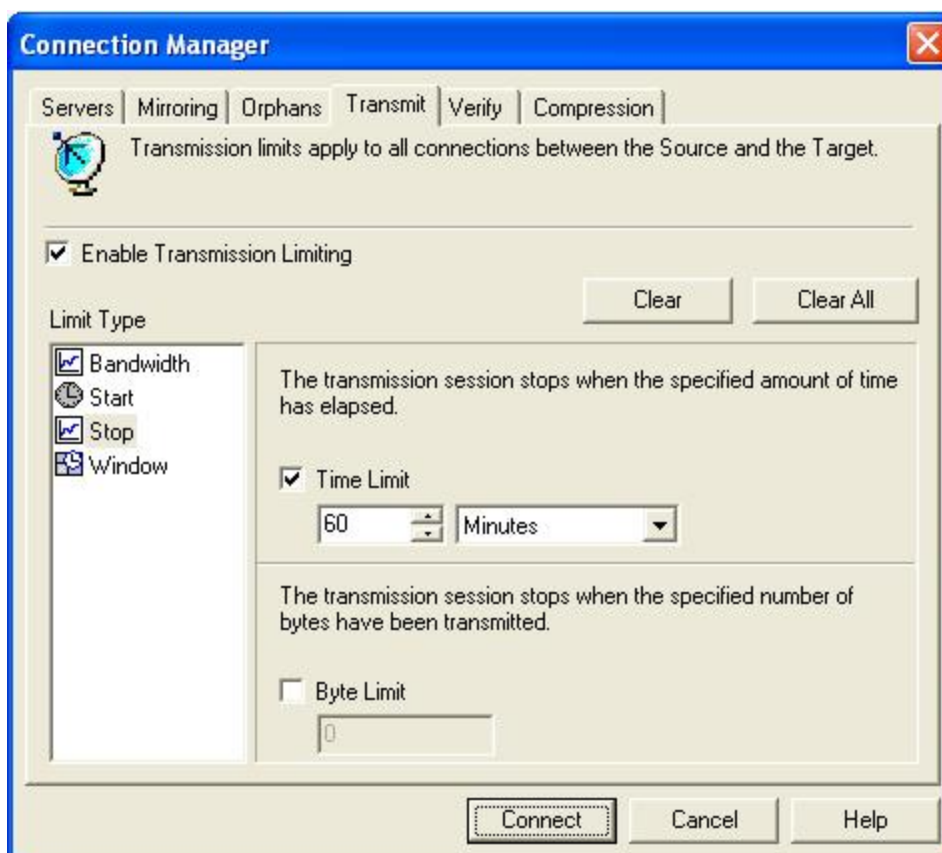
that must be in use to initiate the transmission process.



A **Transmission Session Start** setting will override any other start criteria. For example, if you set the **Transmission Session Start** and the **Queue Threshold**, transmission will not start until you reach the indicated start time.

- **Queue Threshold (bytes)**—This feature is not currently functional.
4. Schedule any desired stop criteria to stop transmission after a transmission start criteria has initiated the transmission. If you do not establish a stop criteria, transmission will end when the queue is empty. Select the **Stop** option in the **Limit Type** box.

Define the stop options to stop Carbonite Availability transmissions by using either or both of the following options.



- **Time Limit**—The time limit specifies the maximum length of time for each transmission period. Any data that is not sent during the specified time limit remains on the source queue. When used in conjunction with the session interval start option, you can explicitly define how often data is transmitted and how long each transmission lasts. Specify the maximum length of time that Carbonite Availability can continue transmitting by indicating a length of time and choosing minutes, hours, or days.
- **Byte Limit**—The byte limit specifies the maximum number of bytes that can be sent before ending the transmission session. When the byte limit is met, Carbonite Availability will automatically stop transmitting data to the target. Any data that still remains waits in the

source queue until the transmission is restarted. When used in conjunction with a session start option, you can explicitly define how much data is being sent at a given time. Specify the maximum number of bytes that can be sent before ending the Carbonite Availability transmission.



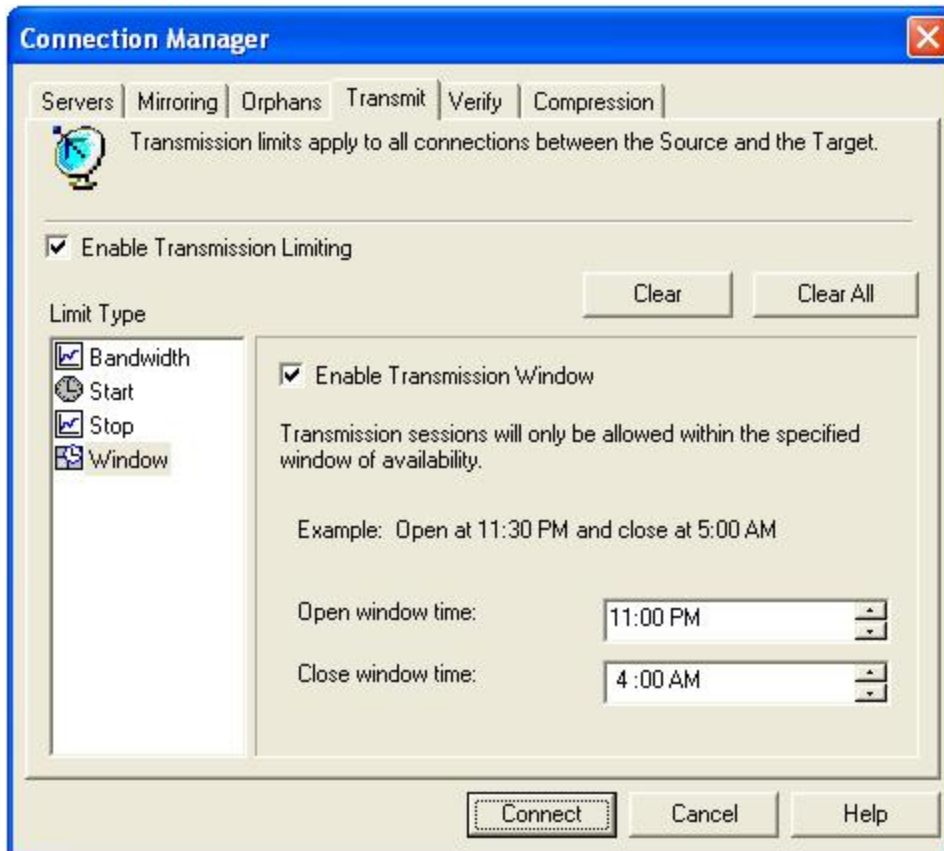
The transmission start and stop criteria should be used in conjunction with each other. For example, if you set the **Queue Threshold** equal to 10 MB and the **Byte Limit** equal to 10 MB, a network connection will be established when there is 10 MB of data in the queue. The data will be transmitted and when the 10 MB **Byte Limit** is reached, the network connection closes. This is useful in configurations where metered charges are based on connection time.

5. Schedule a transmission window to establish a period of availability for all Carbonite Availability transmissions. You can specify a begin and end time for all Carbonite Availability transmissions. When a transmission window is in effect, all other start and stop criteria are bound by this window. This means that Carbonite Availability will never transmit data outside of an established window, regardless of other transmission settings. For example, if you set a window of availability from 9 p.m. to 4 a.m. and a start option to initiate transmission at 5 a.m., the window option will override the start option and no data will be sent at 5 a.m. Select the **Window** option in the **Limit Type** box.
-



Setting a transmission window by itself is not sufficient to start a transmission. You still need to set a start criteria within the window.

Define a window to control Carbonite Availability transmissions by enabling the feature and then specifying both window options.



- **Enable Transmission Window**—This option specifies whether a transmission window is in use.
- **Open window time**—Specifies the time, formatted for AM or PM, when the transmission window will open, allowing transmission to begin.
- **Close window time**—Specifies the time, formatted for AM or PM, when the transmission window will close, stopping all transmission.

6. Click **OK** to save the settings.



When you schedule start criteria for transmission, you may see the transmission status in an error state at the scheduled start. The transmission will still continue as expected.

Limiting transmission bandwidth

Using the Connection Manager **Transmit** tab, you can set start and stop criteria along with a schedule window.



Carbonite Availability checks the schedule once every second, and if a user-defined criteria is met, transmission will start or stop, depending on the option specified.

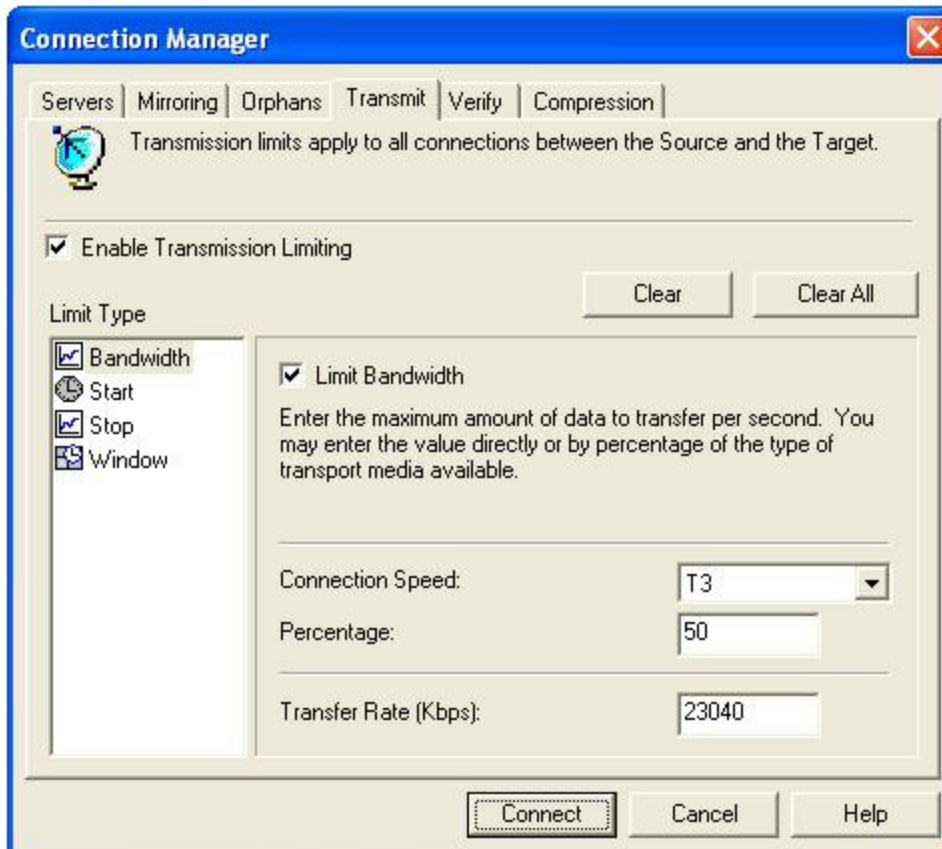
Any replication sets from a source connected to the same IP address on a target will share the same scheduled transmission configuration.

1. Right-click the connection on the right pane of the Replication Console for Linux and select **Connection Manager**.
2. Select the **Transmit** tab. The **Transmit** tab contains four limit types: **Bandwidth**, **Start**, **Stop**, and **Window**. The transmission options for each limit type are displayed by highlighting a selection in the **Limit Type** box.

At the top of the **Transmit** tab dialog box, the **Enable Transmission Limiting** check box allows you to turn the transmission options on or off. You can enable the transmission options by marking the **Enable Transmission Limiting** check box when you want the options to be applied, but you can disable the transmission options, without losing the settings, by clearing that check box.

Also at the top of the **Transmit** tab dialog box, the **Clear All** button, when selected, will remove all transmission limitations that have been set under any of the limit types. The **Clear** button will clear the settings only for the **Limit Type** selected.

3. Select the **Bandwidth** option in the **Limit Type** box. Mark the **Limit Bandwidth** check box to enable the bandwidth limiting features. Define the bandwidth available for Carbonite Availability transmission by using either of the following options.



- **Percentage**—Specify the percentage of bandwidth to be used for Carbonite Availability transmissions and the total bandwidth capacity that is available.
- **Transfer Rate**—Specify the number of kilobits to send every second.



The only value that is persistently stored is the number of kilobits per second. When the page is refreshed, the percentage and available bandwidth capacity may not be the same value that you entered. Carbonite Availability changes these values to the maximum values for the smallest possible link.

4. Click **OK** to save the settings.

Compressing data for transmission

To help reduce the amount of bandwidth needed to transmit Carbonite Availability data, compression allows you to compress data prior to transmitting it across the network. In a WAN environment this provides optimal use of your network resources. If compression is enabled, the data is compressed before it is transmitted from the source. When the target receives the compressed data, it decompresses it and then writes it to disk. On a default Carbonite Availability installation, compression is disabled.



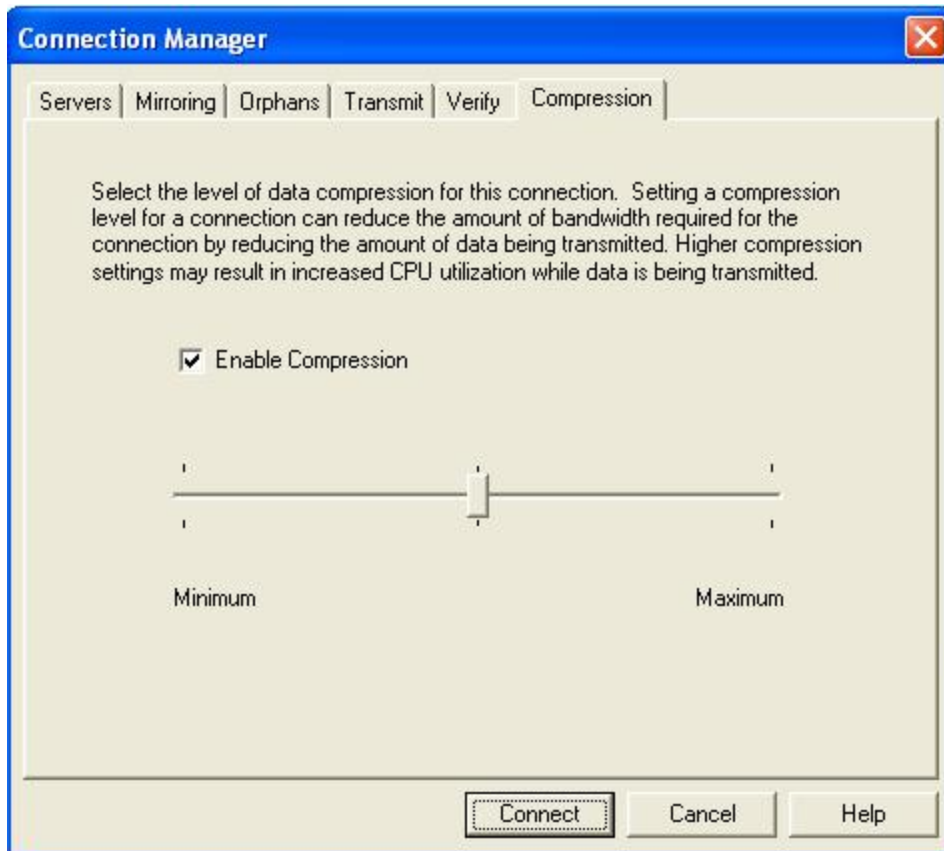
Any replication sets from a source connected to the same IP address on a target will share the same compression configuration.

Keep in mind that the process of compressing data impacts processor usage on the source. If you notice an impact on performance while compression is enabled in your environment, either adjust to a lower level of compression, or leave compression disabled. Use the following guidelines to determine whether you should enable compression:

- If data is being queued on the source at any time, consider enabling compression.
- If the server CPU utilization is averaging over 85%, be cautious about enabling compression.
- The higher the level of compression, the higher the CPU utilization will be.
- Do not enable compression if most of the data is inherently compressed. Many image (.jpg, .gif) and media (.wmv, .mp3, .mpg) files, for example, are already compressed. Some images files, such as .bmp and .tif, are uncompressed, so enabling compression would be beneficial for those types.
- Compression may improve performance even in high-bandwidth environments.
- Do not enable compression in conjunction with a WAN Accelerator. Use one or the other to compress Carbonite Availability data.

Use the following instructions for setting compression.

1. Right-click the connection on the right pane of the Replication Console for Linux and select Connection Manager.
2. Select the **Compression** tab.



3. By default, compression is disabled. To enable it, select **Enable Compression**.
4. Depending on the compression algorithms available for your operating system, you may see a slider bar indicating different compression levels. Set the level from minimum to maximum compression to suit your needs.
5. Click **OK** to save the settings.

Failover

Failover is the process in which a target stands in for a failed source. As a result, user and application requests that are directed to the failed source are routed to the target.

Carbonite Availability monitors the source status by tracking network requests and responses exchanged between the source and target. When a monitored source misses a user-defined number of requests, Carbonite Availability assumes that the machine has failed. Carbonite Availability then prompts the network administrator to initiate failover, or, if configured, it occurs automatically.

The failover target assumes the network identity of the failed source. When the target assumes the identity of the source, user and application requests destined for the source machine or its IP address (es) are routed to the target.

When partnered with the Carbonite Availability data replication capabilities, failover routes user and application requests with minimal disruption and little or no data loss. In some cases, failover may be used without data replication to ensure high availability on a machine that only provides processing services, such as a web server.

Failover can be configured to stand in for one or more IP addresses associated with different NICs on the source. Each IP address can be added to a specific target NIC making NIC configuration very flexible. For example, a single NIC on the source may have one or more IP addresses assigned to it. If that source or the NIC fails, all traffic from the source is directed to the target. If there are multiple NICs on the source, the target can assume the traffic from all of the addresses. Additional NICs on the target increase flexibility and control. Secondary target NICs can assume the traffic from a failed source NIC while normal target traffic can continue to use the primary target NIC.

Failback is the process in which the target releases the source identity so that the source can be brought back onto the network.

- *Configuring failover monitoring* on page 155
- *Editing failover monitoring configuration* on page 162
- *Monitoring failover monitoring* on page 163
- *Failing over* on page 166
- *Removing failover monitoring configuration* on page 166

Configuring failover monitoring

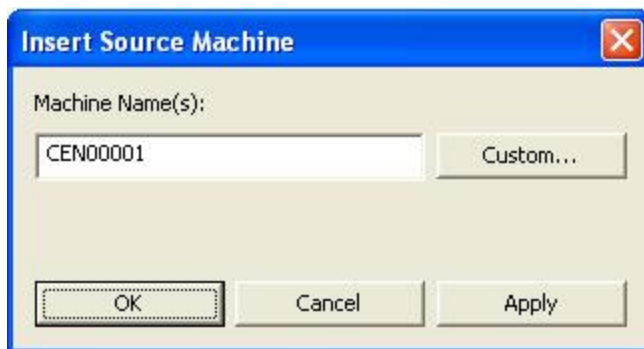
Before beginning your failover configuration, review your IP address and subnet configuration on the source. Because of limitations in the way the Linux kernel handles IP address aliases, you will not be able to mix subnets on the eth0 network interface. Failover should not cause problems in this configuration, but you will lose IP addresses during failback. Therefore, if you must mix subnets on a single interface, use eth1 or higher.

1. The Failover for Linux console can be started from within the Replication Console for Linux or from the Windows desktop.
 - From the Replication Console for Linux, select **Tools, Failover Control Center**.
 - From the Windows desktop, selecting **Carbonite, Replication, Carbonite Failover for Linux** from your **Programs, All Programs, or Apps**, depending on your operating system.
2. Select a failover target from the **Target Machine** list box.



If the target you need is not listed, click **Add Target** and manually enter a name or IP address (with or without a port number). You can also select the **Browse** button to search for a target machine name. Click **OK** to select the target machine and return to the Failover for Linux console main window.

3. Click **Login** to login to the selected target.
4. Select a source machine to monitor by clicking **Add Monitor**. The Insert Source Machine dialog box appears in front of the Monitor Settings dialog box.
5. On the Insert Source Machine dialog, specify your source machine by either of the following methods.



- Type the name of the machine that you want to monitor in **Machine Name(s)** and click **OK**.
- Click **Custom**. Enter the name of the server and click **Add**. Specify the IP address and subnet mask of the specified server and click **OK**. Click **OK** again.

The Insert Source Machine dialog closes and the Monitor Settings dialog remains open with your source listed in the **Names to Monitor** tree.

6. In the **Names to Monitor** tree, locate and select the IP addresses on the source that you want to monitor.
7. Highlight an IP address that you have selected for monitoring and select a **Target Adapter** that

will assume that IP address during failover. Repeat this process for each IP address that is being monitored.

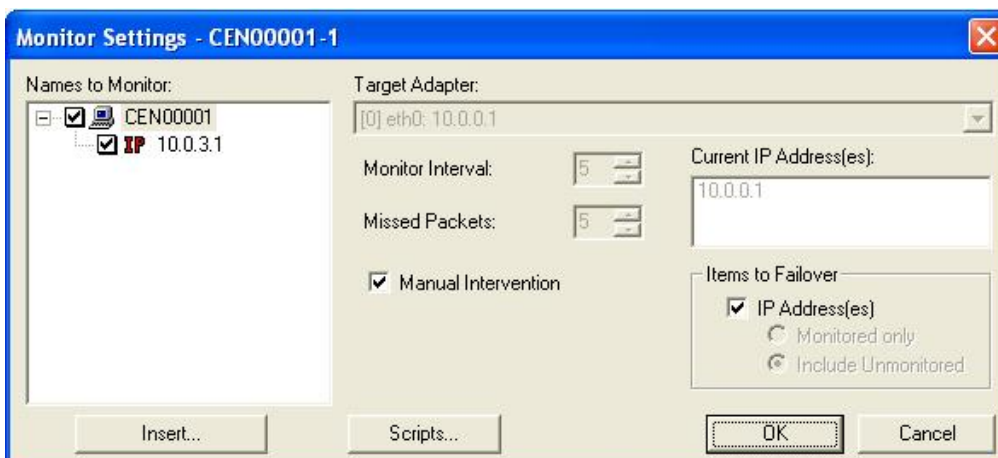


Current IP Addresses displays the IP address(es) currently assigned to the selected target adapter.

8. Highlight an IP address that you have selected for monitoring and select a **Monitor Interval**. This setting identifies the number of seconds between the monitor requests sent from the target to the source to determine if the source is online. Repeat this step for each IP address that is being monitored.
9. Highlight an IP address that you have selected for monitoring and select the **Missed Packets**. This setting is the number of monitor replies sent from the source to the target that can be missed before assuming the source machine has failed. Repeat this step for each IP address that is being monitored.



To achieve shorter delays before failover, use lower **Monitor Interval** and **Missed Packets** values. This may be necessary for IP addresses on machines, such as a web server or order processing database, which must remain available and responsive at all times. Lower values should be used where redundant interfaces and high-speed, reliable network links are available to prevent the false detection of failure. If the hardware does not support reliable communications, lower values can lead to premature failover. To achieve longer delays before failover, choose higher values. This may be necessary for IP addresses on slower networks or on a server that is not transaction critical. For example, failover would not be necessary in the case of a server restart.



10. Highlight the source name and specify the **Items to Failover**, which identifies which source components you want to failover to the target.
 - **IP Addresses**—If you want to failover the IP addresses on the source, enable this option and then specify the addresses that you want to failover. When the source and target are on the same subnet, generally a LAN environment, you should failover the IP address. If the source and target are on different subnets, generally a WAN environment, you should not failover the IP address. See *WAN considerations* on page 158 for options on handling WAN failover.

- **Monitored only**—Only the IP address(es) that are selected for monitoring will be failed over.
 - **Include Unmonitored**—All of the IP address(es) will be failed over.
-



If you are monitoring multiple IP addresses, IP address conflicts may occur during failover when the number of IP addresses that trigger failover is less than the number of IP addresses that are assumed by the target during failover. For example, if a source has four IP addresses (three public and one private), and two of the three public addresses are monitored, but all three public addresses are configured to failover, a conflict could occur. If the source fails, there is no conflict because all of the IP addresses have failed and no longer exist. But if the failure only occurs on one of the monitored addresses, the other two IP addresses are still affected. If all of the addresses are failed over, these addresses then exist on both the source and the target. Therefore, when a source machine has fewer IP addresses that trigger failover than IP addresses that will be failed over, there is a risk of an IP address conflict.

11. By default, **Manual Intervention** is enabled, allowing you to control when failover occurs. When a failure occurs, a prompt appears in the Failover for Linux console and waits for you to manually initiate the failover process. Disable this option only if you want failover to occur immediately when a failure occurs.
12. If you are using any failover or failback scripts, click **Scripts** and enter the path and filename for each script type. Scripts may contain any valid Linux command, executable, or script file. Examples of functions specified in scripts include stopping services on the target before failover because they may not be necessary while the target is standing in for the source, stopping services on the target that need to be restarted with the source's machine name and IP address, starting services or loading applications that are in an idle, standby mode waiting for failover to occur, notifying the administrator before and after failover or failback occurs, stopping services on the target after failback because they are no longer needed, stopping services on the target that need to be restarted with the target machine's original name and IP address, and so on. Specify each script that you want to run and the following options, if necessary.
13. If you want to delay the failover or failback processes until the associated script has completed, mark the appropriate check box.
14. If you want the same scripts to be used as the default for future monitor sessions, mark the appropriate check box.
15. Click **OK** to return to the Monitor Settings dialog box.
16. Click **OK** on the Monitor Settings dialog box to save your monitor settings and begin monitoring for a failure.

WAN considerations

When the source and target are on the same subnet, generally a LAN environment, you should failover the IP address. However, if the source and target are on different subnets, generally a WAN environment, you should not failover the IP address. You have several options for handling WAN failover.

- **DNS updates**—You can script DNS updates to modify, at failover time, the source server's DNS A records to have the IP address of the target. When clients resolve a name to an IP address, they will resolve to the target IP address. Depending on the domain size and how DNS updates are propagated, it may take several minutes or even hours for the updates to complete.
- **Reconfigure routers using a failover script**—You can automatically reconfigure routers using a failover script to move the source's subnet from the source's physical network to the target's physical network. Since the route to the source's subnet will be changed at failover, the source server must be the only system on that subnet, which in turn requires all server communications to pass through a router. Additionally, it may take several minutes or even hours for routing tables on other routers throughout the network to converge.
- **VPN infrastructure**—A VPN infrastructure allows your source and target to be on the same subnet, in which case IP address failover will work the same as a LAN configuration.

Scripting example using the BIND DNS client

1. Install the BIND DNS client on the target server, if it is not already installed.
2. Create a PATH statement on the target for the BIND directory to ensure that it runs every time the executable is called.
3. Update the Carbonite Availability service on the target to use a domain account that has rights to modify BIND DNS. You will have to stop and restart the service for changes to the user account to take effect.
4. Create a failover script with the following command. Specify this script for post-failover.

```
nsupdate "dnsover"
```

5. Create a file called dnsover and add the following lines. This is the file called by your post-failover script.

```
# Substitute your source name, target name, and target IP address
update delete source_server_name.fully_qualified_domain.com
update add target_server_name.fully_qualified_domain.com 86400 A target_server_IP
send
```

6. Create a failback script with the following command. Specify this script for post-failback.

```
nsupdate "dnsback"
```

7. Create a file called dnsback and add the following lines. This is the file called by your post-failback script.

```
# Substitute your target name, source name, and source IP address
update delete target_server_name.fully_qualified_domain.com A
update add source_server_name.fully_qualified_domain.com 86400 A source_server_IP
send
```

When failover and failback occur, the failover and failback scripts will automatically trigger DNS updates.

Protecting NFS exports

NFS exports must be configured for failover through the failover scripts or created manually on the target after failover.

1. Start the Carbonite Availability service on the source.
2. Stop and restart the NFS service on the source. The Carbonite Availability service must be running before the NFS service in order for replication operations to be captured.
3. On your target, set the NFS service to manual startup. This allows the failover script to control when the service starts on the target.
4. Create a replication set on the source that includes `/etc/exports` and the shared data.
5. Connect the replication set using the Connection Wizard or the Connection Manager.
6. Add the following to your post-failover script.

```
service nfs start
```

7. If necessary, update DNS.

After failover, the NFS service will automatically be started by the post-failover script. If your clients see a stale file handle error message when attempting to access an export, they will need to reconnect to it.

Protecting Samba shares

A share is any local volume, drive, or directory resource that is exported and shared across a network. Samba shares must be configured for failover through the failover scripts or created manually on the target.

1. Start the Carbonite Availability service on the source.
2. Stop and restart the Samba service on the source. The Carbonite Availability service must be running before the Samba service in order for replication operations to be captured.
3. On your target, set the SMB and WinBind services to manual startup. This allows the failover script to control when the services start on the target.
4. Create a replication set on the source that includes `/etc/SAMBA/samba_conf` and the shared data.
5. Connect the replication set using the Connection Wizard or the Connection Manager.
6. Add the following to your post-failover script.

```
service smb start
```

7. If necessary, update DNS.

After failover, the services will automatically be started by the post-failover script. If your clients see an access denied or share not found error message when attempting to access a share, they will need to remount the share.

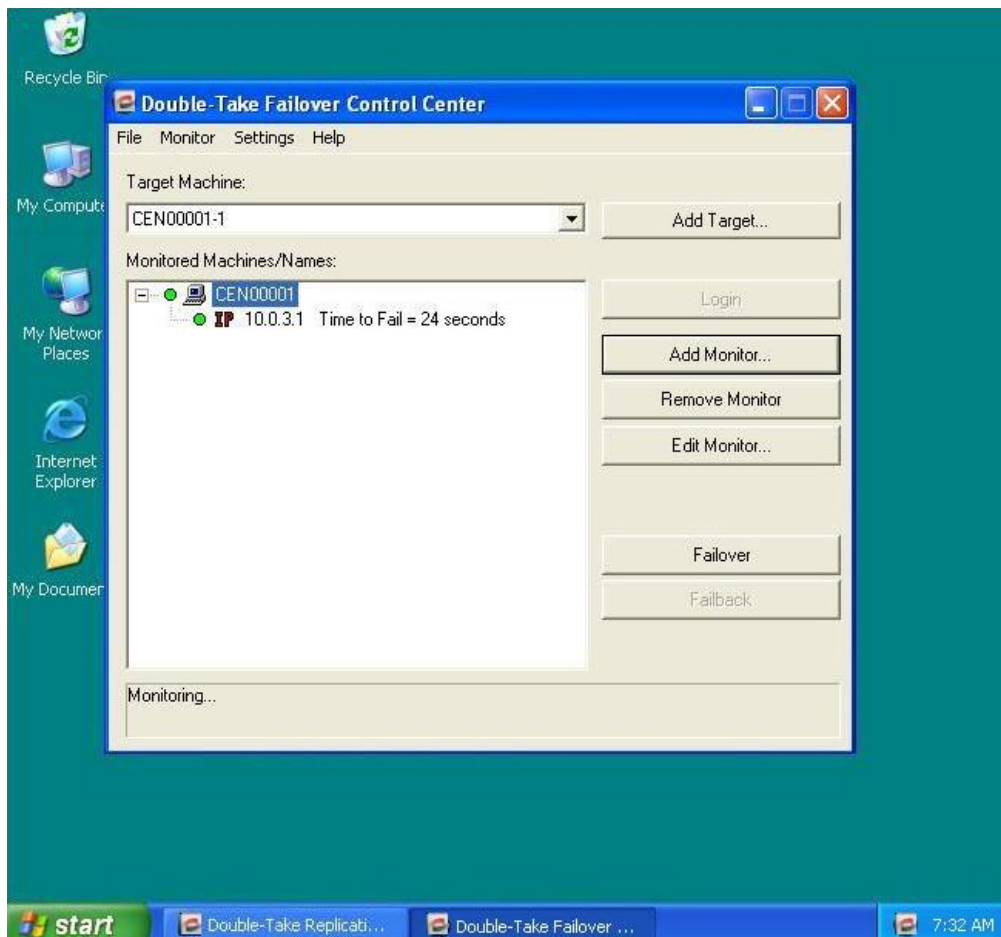
Editing failover monitoring configuration

If you want to edit the monitor settings for a source that is currently being monitored, highlight that source on the Monitored Machines tree on the main Failover for Linux console screen and click **Edit**. The Monitor Settings dialog box will open. See *Configuring failover monitoring* on page 155.

Monitoring failover monitoring

Since it can be essential to quickly know the status of failover monitoring, Carbonite Availability offers various methods for monitoring failover monitoring. When the Failover for Linux console is running, you will see four visual indicators:

- The Failover for Linux console Time to Fail counter
- The Failover for Linux console status bar located at the bottom of the window
- The Failover for Linux console colored bullets to the left of each IP address and source machine
- The Windows desktop icon tray containing a failover icon



You can minimize the Failover for Linux console and, although it will not appear in your Windows taskbar, it will still be active and the failover icon will still appear in the desktop icon tray.

The Failover for Linux console does not have to be running for failover to occur.

The following table identifies how the visual indicators change when the source is online.

Time to Fail Countdown

The Time to Fail counter is counting down and resetting each time a response is received from the source machine.

Status Bar

The status bar indicates that the target machine is monitoring the source machine.

Colored Bullets

The bullets are green.

When the Time to Fail value has decreased by 25% of the entire timeout period, the bullet changes from green to yellow, indicating that the target has not received a response from the source. The yellow bullet is a caution signal. If a response from the source is received, the countdown resets and the bullets change back to green. If the countdown reaches zero without the target receiving a response from the source, failover begins.

Desktop Icon Tray

The Windows desktop icon tray contains a failover icon with red and green computers.

The following table identifies how the visual indicators change when the source fails and failover is initiated.

Time to Fail Countdown

The Time to Fail countdown value is 0.

Status Bar

The status bar displays the source machine and IP address currently being assumed by the target.

Colored Bullets

The bullets are red.

Desktop Icon Tray

The Windows desktop icon tray contains a failover icon with red and green computers.

The following table identifies how the visual indicators change when failover is complete.

Time to Fail Countdown

The Time to Fail counter is replaced with a failed message.

Status Bar

The status bar indicates that monitoring has continued.

Colored Bullets

The bullets are red.

Desktop Icon Tray

The Windows desktop icon tray contains a failover icon with a red computer.

Failing over

The failover process, including script processing, can be tested at any time. To force unavailability, disconnect the network cable from a monitored machine, wait for the **Time to Fail** counter to decrease to zero and failover begins. To avoid the countdown delay, highlight the monitored machine name in the Failover for Linux console window and select **Failover**.

If **Manual Intervention** is enabled, the Failover for Linux console will prompt you when a failure occurs.



If the Failover for Linux console is not running at the time the failure occurs, the manual intervention dialog box will appear the next time the Failover for Linux console is started.

When a failure occurs, an alert is forwarded to the Linux system log. You can then start the Failover for Linux console and respond to the manual intervention prompt.

If SNMP is installed and configured, an SNMP trap is also generated. When using a third-party SNMP manager, an e-mail or page can be generated to notify you of the failure.

Files that were open or being accessed at the time of failover will generate Stale NFS file handle error messages. Remount the NFS export to correct this error.

Click **Cancel** to abort the failover process. If necessary, you can initiate failover later from the Failover for Linux console. Click **OK** to proceed with failover.

Removing failover monitoring configuration

If you want to discontinue monitoring a source, highlight that machine on the Monitored Machines tree on the main Failover for Linux console screen and click **Remove Monitor**. No additional dialog boxes will open.

Failback and restoration

Failover occurred because the target was monitoring the source for a failure, and when a failure occurred, the target stood in for the source. User and application requests that were directed to the failed source are routed to the target.

While the users are accessing their data on the target, you can repair the issue(s) on the source. Before users can access the source again, you will need to restore the data from the target back to the source and perform failback. Failback is the process where the target releases the source identity it assumed during failover. Once failback is complete, user and application requests are no longer routed to the target, but back to the source.

Ideally, you want to restore your data from the target back to the source before you failback. This allows users who are currently accessing their data on the target because of failover to continue accessing their data. Restoration before failback reduces user downtime. The other method allows you to failback first and then restore the data from the target to the source. This method may be easier in some situations, but users may experience longer downtime, depending on the amount of data to be restored, because they will be unable to access their data during both the restoration and the failback processes.

- *Restoring then failing back* on page 168
- *Failing back then restoring* on page 173

Restoring then failing back

Use these instructions to restore your data first and then failback.

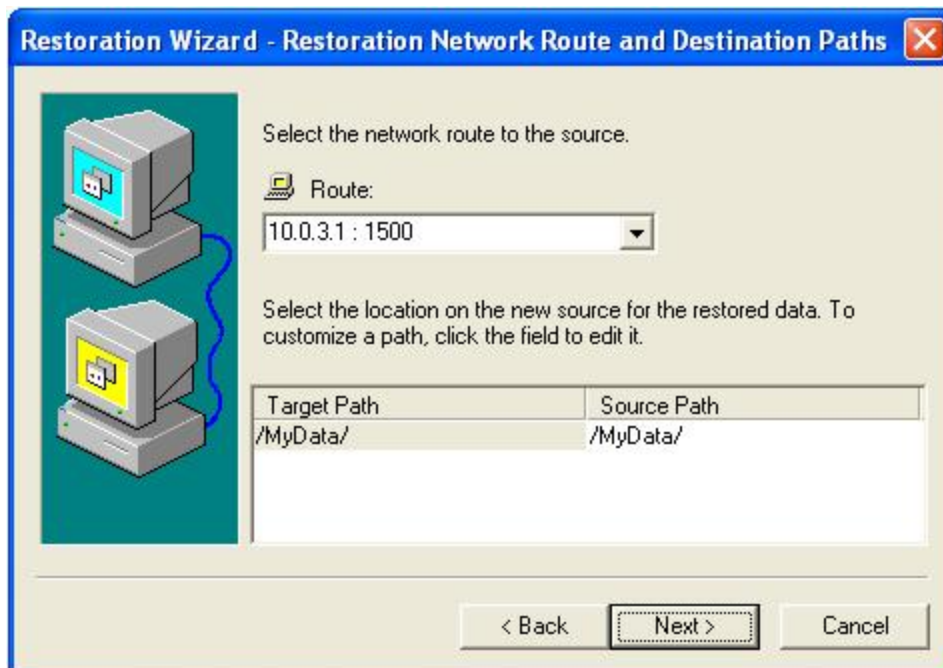
1. Resolve the problem(s) on the source that caused it to fail. If you have to rebuild your source, use a unique identity.
2. Stop any applications that were failed over that may be running on your source. The files must be closed on the source so that updated files from the target will successfully overwrite the files on the source during the restoration.
3. Complete the following steps on your source to prevent Carbonite Availability from automatically reconnecting any connections.
 - a. Stop the Double-Take service.
 - b. Rename `/var/lib/DT/connect.sts` to a unique file name.
 - c. Start the Double-Take service.
4. Modify the source so that it can be brought onto the network with a new, unique IP address or one that was not failed over. It needs to be able to exist on the network without an IP address conflict and communicate with the target.
5. At this point, confirm you have the following configuration.
 - Your target is standing in for your source because of failover, and users are accessing their data from the target.
 - Your source is back online with a unique IP address.
 - The source and target can communicate with each other.
 - All applications on the source are stopped.
6. From your target, confirm the Replication Console for Linux is communicating with the source using the new, unique IP address.
 - a. From the Replication Console for Linux on the target, right-click the source and select **Remove**.
 - b. Depending on your configuration, the source may be automatically inserted back into the Replication Console for Linux. If it is not, select **Insert, Server**. Specify the source server by the new IP address and click **OK**.
7. Begin your restoration process.
 - a. From the Replication Console for Linux, select **Tools, Restoration Wizard**.
 - b. Review the Welcome screen and click **Next** to continue.



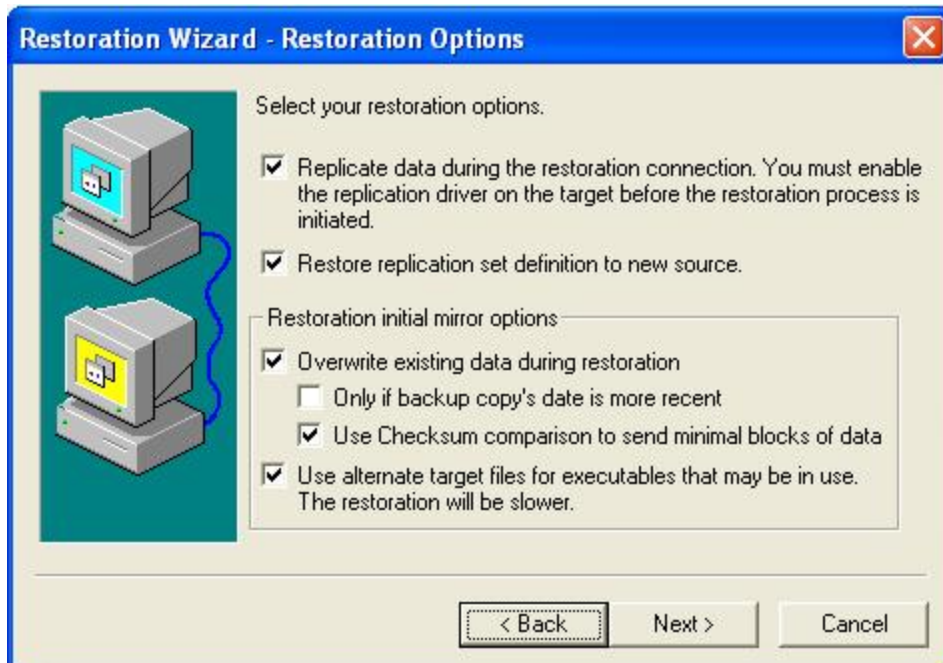
At any time while using the Restoration Wizard, click **Back** to return to previous screens and review your selections.

- c. Select the target that contains the current copy of the data that you want to restore and click **Next**.
- d. Select the original source or **Alternate**, if your original source is not listed. This option identifies to the target which data set you are trying to restore so that the appropriate replication sets can be presented to you.
- e. Click **Next** to continue.

- f. Specify if you want to use an existing replication set or create a new one. This replication set will be used to connect from the target to the source.
 - **Use this replication set**—If you choose to use an existing replication set, specify the name of that replication set by selecting it from the pull-down menu. You will have an opportunity to modify the replication set definition.
 - **Create a new replication set with this name**—If you choose to create a new replication set, specify a replication set name. With this option, you will need to define the data to be restored.
- g. Click **Next** to continue.
- h. A tree display appears identifying the data available for restoration. Mark the check box of the volumes, directories, and/or files you want to restore. Keep in mind that if you exclude volumes, folders, and/or files that were originally replicated, it may compromise the integrity of your applications or data.
- i. Click **Next** to continue
- j. Select the new source server. This is the server where the data from the target will be restored. This may be the original source server or a new server. Click **Next** to continue.
- k. Select your network route to the new source, which includes the IP address and port number. Also select the location on the new source for the restored data. If you want to set a customized path, click in the field under **Source Path** to edit the location.



- l. Click **Next** to continue.
- m. Specify the restoration options that you want to use.



- **Replicate data during the restoration connection**—This option allows you to replicate on-going data changes during and after the restoration mirror is performed. Use this option if the source data on the target will continue to change during the restoration process. You do not need to use this option if the source data on the target is not changing. If you do not select this option, any data changes that might occur on the target after the restoration process is initiated will not be transmitted to the source. If you do select this option, you must configure replication on the target prior to initiating the restoration process.
- **Restore replication set definition to new source**—This option restores a copy of the replication set database on the target to the new source.
- **Overwrite existing data during restore**—This option restores all existing files by overwriting them and writes any files that do not exist. If this option is disabled, only files that do not exist on the new source will be restored.
 - **Only if backup copy's date is more recent**—This option restores only those files that are newer on the target than on the new source. The entire file is overwritten with this option.



If you are using a database application, do not use the newer option unless you know for certain you need it. With database applications, it is critical that all files, not just some of them that might be newer, get restored.

-
- **Use Checksum comparison to send minimal blocks of data**—Specify if you want the restoration process to use a block checksum comparison to determine which blocks are different. If this option is enabled, only those blocks (not the entire files) that are different will be restored to the new source.



To ensure data integrity, the replicate during restoration and overwrite existing data options are dependent on each other. If you want to enable replication, overwrite data will automatically be enabled. If you disable the option to overwrite data, replication will automatically be disabled.

- **Use alternate target files for executables that may be in use**—If you have executables that may be in use during the restoration, you can have Carbonite Availability create and update an alternate file during the restoration. Once the mirroring and replication operations have been completed, the alternate file will be renamed to the original file. This process will reduce the speed of your restoration, so it should only be used if executables may be in use.
- n. Review your selections and click **Finish** to begin the restoration.
 8. Monitoring the restoration connection and after the **Mirror Status** is **Idle**, schedule a time for failback. User downtime will begin once failback is started, so select a time that will have minimal disruption on your users.
 9. When you are ready, begin the failback process.
 - a. Stop user access to the target. The user downtime starts now.
 - b. In the Replication Console for Linux, watch the restoration connection until activity has ended and replication is in a **Ready** state. This will happen as the final data in queue, if any, is applied on the source. The replication **Ready** state indicates replication is waiting for new incoming data changes.
 - c. Disconnect the restoration connection.
 - d. Open the Failover for Linux console.
 - e. Select the original target that is currently standing in for the original failed source.
 - f. Highlight the failed source and click **Failback**. If you have a pre-failback script configured, it will be started.
 - g. When failback is complete, the post-failback script, if configured, will be started. When the script is complete, you will be prompted to determine if you want to continue failover monitoring, do not select either option. Leave the prompt dialog box open as is.
 10. Complete the following steps on your source to update the source to its original identity.
 - a. Stop the Double-Take service.
 - b. Modify the source identity back to the original source IP address.
 - c. Start the Double-Take service.
 11. Confirm the Replication Console for Linux is communicating with the source using the original IP address.
 - a. Right-click the source and select **Remove**.
 - b. Depending on your configuration, the source may be automatically inserted back into the Replication Console for Linux. If it is not, select **Insert, Server**. Specify the source server by the original IP address and click **OK**.
 12. At this time, you can go back to the dialog box in the Failover for Linux console. Select **Continue** or **Stop** to indicate if you want to continue monitoring the source. After you have selected whether or not to continue monitoring the source, the source post-failback script, if configured, will be started.



The source must be online and Carbonite Availability must be running to ensure that the source post-failback script can be started. If the source has not completed its boot process, the command to start the script may be lost and the script will not be initiated.

At this time, you can start any applications and allow end-users to access the data.

Failing back then restoring

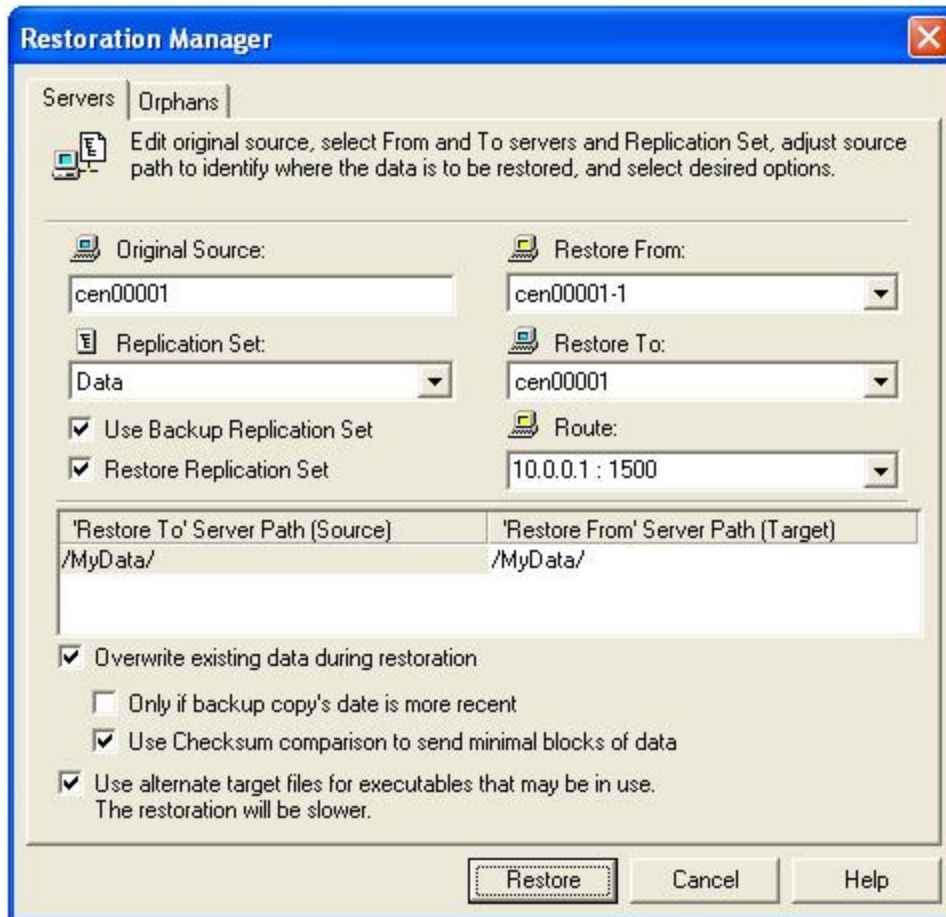
Use these instructions to failback first and then restore your data.

1. Resolve the problem(s) on the source that caused it to fail. If you have to rebuild your source, make sure you use the same identity as the original source configuration.
2. Because you do not want your users accessing the source or its data until newer data from the target can be restored, deny access to user logins by setting `/etc/nologin`. See your Linux documentation for details on creating this file.
3. Stop any applications that may be running on your source. The files must be closed on the source so that updated files from the target will overwrite the files on the source during the restoration.
4. From the Failover for Linux console, select the target that is currently standing in for the failed source.
5. Select the failed source and click **Failback**. The user downtime starts now. If you have a pre-failback script configured, it will be started.
6. When failback is complete, the post-failback script, if configured, will be started. When the script is complete, you will be prompted to determine if you want to continue monitoring the source. Select **Continue** or **Stop** to indicate if you want to continue monitoring the source. After you have selected whether or not to continue monitoring the source machine, the source post-failback script, if configured, will be started.



The source must be online and Carbonite Availability must be running to ensure that the source post-failback script can be started. If the source has not completed its boot process, the command to start the script may be lost and the script will not be initiated.

7. From the Replication Console for Linux, select **Tools, Restoration Manager**.



8. Identify the **Original Source** machine. This is your source machine where the data originally resided.
9. Select the **Restore From** machine. This is the target machine where the copy of the data is stored.
10. **Replication Set** contains the replication set information stored on the target machine (the machine in **Restore From**). If no replication sets are available, the list will be blank. Select the replication set that corresponds to the data that you need to restore.
11. Select the **Restore To** machine. This is your source where the updated data from the target will be sent.
12. Select the **Use Backup Replication Set** check box to use the target's copy of the replication set database for the restoration. If this check box is not marked, you will be accessing the replication set information from the source.
13. Select the **Restore Replication Set** check box to restore the target's copy of the replication set database to the source during the restoration process.
14. Select the **Route** on the target. This is the IP address and port on the target that the data will be transmitted through. This allows you to select a different route for Carbonite Availability traffic. For example, you can separate regular network traffic and Carbonite Availability traffic on a machine with multiple IP addresses.
15. The **Restore To Server Path** and **Restore From Server Path** paths will automatically be populated when the replication set is selected. The restore to path is the directory that is the

common parent directory for all of the directories in the replication set. If the replication set crosses volumes, then there will be a separate path for each volume. The restore from path is the path on the target server where the replicated files are located.



Restoring across a NAT router requires the ports to be the same as the original connection. If the ports have been modified (manually or reinstalled), you must set the port numbers to the same values as the last valid source/target connection.

16. Select the restoration conditionals that you want to use.

- **Overwrite existing data during restore**—This option restores all existing files by overwriting them. Any files that do not exist on the source are written also. If this option is disabled, only files that do not exist on the source will be restored.
 - **Only if backup copy's date is more recent**—This option restores only those files that are newer on the target than on the source. The entire file is overwritten with this option.
-



If you are using a database application, do not use the newer option unless you know for certain you need it. With database applications, it is critical that all files, not just some of them that might be newer, get mirrored.

- **Use Checksum comparison to send minimal blocks of data**—Specify if you want the restoration process to use a block checksum comparison to determine which blocks are different. If this option is enabled, only those blocks (not the entire files) that are different will be restored to the source.
17. If you want to configure orphan files, click the **Orphans** tab. The same orphan options are available for a restoration connection as a standard connection.
18. Click **Restore** to begin the restoration.

After the restoration is complete, the restoration connection will automatically be disconnected and the replication set deleted. At this time, you can start any applications and allow end-users to access the data on the source.

Chapter 4 Full server protection

Create a full server job when you want to protect the entire source, including the server's system state. You can also use it to protect an application server. This type of job is the most flexible, allowing you to go from physical to physical, physical to virtual, virtual to virtual, and virtual to physical. For full server protection, you will need to complete the following steps, in order.

1. Review the *Full server requirements* on page 177 to make sure your environment meets the requirements.
2. Install the Carbonite Replication Console on a Windows machine.
3. Install Carbonite Availability on your Linux source and target servers.
4. Add your servers to your Carbonite Replication Console. See *Adding servers* on page 64.
5. Create your Linux full server job. See *Creating a full server job* on page 184.



For installation and licensing instructions, see the *Carbonite Availability and Carbonite Move Installation, Licensing, and Activation* document.

Once your job is created and running, see the following sections to manage your job.

- *Managing and controlling full server jobs* on page 202—You can view status information about your job and learn how to control the job.
- *Failing over full server jobs* on page 219—Use this section when a failover condition has been met or whenever you want to failover.
- *Reversing full server jobs* on page 221—Use this section to reverse protection. The source (what was your original target hardware) is now sending data to the target (what was your original source hardware).

Full server requirements

Use these requirements for Linux full server protection. Keep in mind that a target server may meet these requirements but may not be suitable to stand-in for a source in the event of a source failure. See *Target compatibility* on page 182 for additional information regarding an appropriate target server for your particular source.

- **Source and target servers**—The source and target servers can be a physical or virtual server running any of the following operating systems.
 - **Operating system**—Red Hat Enterprise Linux, Oracle Enterprise Linux, and CentOS
 - **Version**—5.9 through 5.11
 - **Kernel type for x86 (32-bit) architectures**—Default, SMP, Xen, PAE
 - **Kernel type for x86-64 (64-bit) architectures**—Default, SMP, Xen
 - **File system**—Ext3, Ext4, XFS
 - **Notes**—Oracle Enterprise Linux support is for the mainline kernel only, not the Unbreakable kernel.
 - **Operating system**—Red Hat Enterprise Linux, Oracle Enterprise Linux, and CentOS
 - **Version**—6.7 through 6.9
 - **Kernel type for x86 (32-bit) architectures**—Default
 - **Kernel type for x86-64 (64-bit) architectures**—Default
 - **File system**—Ext3, Ext4, XFS (64-bit only)
 - **Operating system**—Red Hat Enterprise Linux, Oracle Enterprise Linux, and CentOS
 - **Version**—7.2 through 7.4
 - **Kernel type for x86 (32-bit) architectures**—No 32-bit architectures are supported
 - **Kernel type for x86-64 (64-bit) architectures**—Default
 - **File system**—Ext3, Ext4, XFS
 - **Notes**—The following caveats apply to version 7.4.
 - For full server jobs, your source and target must both be running version 7.4. You cannot have a source running an operating system earlier than version 7.4 protected to a target running version 7.4.
 - Oracle Enterprise Linux version 7.4 is for the Mainline kernel only.
 - **Operating system**—SUSE Linux Enterprise
 - **Version**—11.2 through 11.4
 - **Kernel type for x86 (32-bit) architectures**—Default, Xen, XenPAE, VMI
 - **Kernel type for x86-64 (64-bit) architectures**—Default, Xen
 - **File system**—Ext3, XFS
 - **Operating system**—SUSE Linux Enterprise
 - **Version**—12.0 through 12.2
 - **Kernel type for x86 (32-bit) architectures**—No 32-bit architectures are supported
 - **Kernel type for x86-64 (64-bit) architectures**—Default

- **File system**—Ext3, Ext4, XFS, Btrfs
- **Notes**—If you are planning to convert an existing file system to Btrfs, you must delete any existing Carbonite Availability jobs and re-create them after converting to Btrfs. Also Btrfs cannot be failed over together with ext4. Btrfs and ext4 can be combined with other file systems but not with each other.
- **Operating system**—Ubuntu
 - **Version**—12.04.3, 12.04.4, and 12.04.5
 - **Kernel type for x86 (32-bit) architectures**—Generic
 - **Kernel type for x86-64 (64-bit) architectures**—Generic
 - **File system**—Ext2, Ext3, Ext4, XFS
- **Operating system**—Ubuntu
 - **Version**—14.04.3, 14.04.4, and 14.04.5
 - **Kernel type for x86 (32-bit) architectures**—Generic
 - **Kernel type for x86-64 (64-bit) architectures**—Generic
 - **File system**—Ext2, Ext3, Ext4, XFS
- **Operating system**—Ubuntu
 - **Version**—16.04.0, 16.04.1, and 16.04.2
 - **Kernel type for x86 (32-bit) architectures**—Generic
 - **Kernel type for x86-64 (64-bit) architectures**—Generic
 - **File system**—Ext2, Ext3, Ext4, XFS



For all operating systems except Ubuntu, the kernel version must match the expected kernel for the specified release version. For example, if `/etc/redhat-release` declares the system to be a Redhat 6.6 system, the kernel that is installed must match that.

Carbonite Availability does not support stacking filesystems, like eCryptFS.

- **Packages and services**—Each Linux server must have the following packages and services installed before you can install and use Carbonite Availability. See your operating system documentation for details on these packages and utilities.
 - sshd (or the package that installs sshd)
 - parted
 - `/usr/bin/which`
 - `/usr/sbin/dmidecode`
 - `/usr/bin/scp` (only if you will be performing push installations from the Carbonite Replication Console to your Linux servers)
 - `/lib/lsb/init-functions` (only if you are installing on Red Hat, Oracle Enterprise Linux, CentOS, or SUSE)
- **Source and target preparation**—Make sure your source and target servers are prepared for mirroring, replication, and failover by following these guidelines.
 - Uninstall any applications or operating system features that are not needed from both your source and target. Ideally, your target should be as clean and simple a configuration as possible.

- Install on the source any drivers that are required on the target after failover. For example, you need to install on the source any NIC drivers that will be required on the target after failover.
- Resolve any maintenance updates on the source that may require the server to be rebooted before failover.
- Do not failover if the target is waiting on a reboot after applying maintenance. If failover occurs before the required reboot, the target may not operate properly or it may not boot.
- **System memory**—The minimum system memory on each server is 1 GB.
- **Disk space for program files**—This is the amount of disk space needed for the Carbonite Availability program files. This is approximately 400 MB on each Linux server.



Make sure you have additional disk space for Carbonite Availability queuing, logging, and so on.

-
- **Server name**—Carbonite Availability includes Unicode file system support, but your server name must still be in ASCII format. Additionally, all Carbonite Availability servers must have a unique server name.
 - **Protocols and networking**—Your servers must meet the following protocol and networking requirements.
 - Your servers must have TCP/IP with static IP addressing.
 - IPv4 is the only supported version.
 - If you are using Carbonite Availability over a WAN and do not have DNS name resolution, you will need to add the host names to the local hosts file on each server running Carbonite Availability.
 - **NAT support**—Carbonite Availability supports IP and port forwarding in NAT environments with the following caveats.
 - Only IPv4 is supported.
 - Only standalone servers are supported.
 - Make sure you have added your server to the Carbonite Replication Console using the correct public or private IP address. The name or IP address you use to add a server to the console is dependent on where you are running the console. Specify the private IP address of any servers on the same side of the router as the console. Specify the public IP address of any servers on the other side of the router as the console.
 - DNS failover and updates will depend on your configuration
 - Only the source or target can be behind a router, not both.
 - The DNS server must be routable from the target
 - **Name resolution**—Your servers must have name resolution or DNS. The Carbonite Replication Console must be able to resolve the target, and the target must be able to resolve all source servers. For details on name resolution options, see your Linux documentation or online Linux resources.
 - **Ports**—Port 1501 is used for localhost communication between the engine and management service and should be opened inbound and outbound for both TCP and UDP in iptables. Ports

1500, 1505, 1506, 6325, and 6326 are used for component communication and must be opened inbound and outbound for both TCP and UDP on any firewall that might be in use.

- **Security**—Carbonite Availability security is granted through membership in user groups. The groups can be local or LDAP (Lightweight Directory Access Protocol). A user must provide a valid local account that is a member of the Carbonite Availability security groups.
- **SELinux policy**—SELinux should be disabled on the source and target.
- **UEFI, trusted boot, secure boot**—UEFI (Unified Extensible Firmware Interface) is supported on the source and target, however, trusted boot (tboot), secure boot, or other volume blocking mechanisms are not supported on the source and target.



If you are using SUSE Linux Enterprise version 11.4, you cannot mix UEFI and BIOS. With this version, the source and target must be the same.

- **Docker**—Your source cannot be a Docker host.
- **Mount option**—The mount option noexec is not supported on the /tmp filesystem.
- **Kernel**—Paravirtualized kernels are not supported on the source and target.
- **VMware Tools**—Any VMWare guest running Carbonite Availability should have the appropriate VMWare Tools package installed.
- **Snapshots**—You can take and failover to snapshots using a full server job. Keep in mind the following caveats.
 - You must have LVM on your source and target server. The snapshots will be stored in the LVM volume group within the same volume group as the parent volume, so make sure you have enough free space to accommodate the snapshots. If you do not have enough free space, Carbonite Availability will delete enough snapshots (typically one) to free space for the new snapshot. Bad or overflow snapshots will be deleted first and if there are none of those, then the oldest snapshot will be deleted.
 - Snapshots of / and /boot are not included, even with LVM on the source.
 - There may be a performance impact if you take a lot of snapshots. The more snapshots you have, the longer it may take to create a new snapshot because the volume write time slows down for every snapshot because every block written could cause a write to each snapshot volume.
 - Snapshots are not supported with Btrfs file systems.
 - Snapshots are not supported on Ubuntu 12.04.x. Once a job is created, you will still see Carbonite Availability functionality to take snapshots, but due to operating system limitations specifically with Ubuntu 12.04.x, the snapshots will not be usable.
- **Supported configurations**—The following table identifies the supported configurations for a full server job.

Server Configuration	Description	Supported	Not Supported
One to one active/standby	You can protect a single source to a single target. The target has no production activity. The source is the only server actively replicating data.	X	
One to one active/active	You cannot protect a single source to a single target where each server acts as both a source and target actively replicating data to each other.		X
Many to one	You cannot protect many source servers to one target server.		X
One to many	You can protect a single source to multiple target servers. The source is the only server actively replicating data. This will create redundant copies of your source. You will only be able to configure reverse protection for the first job. Subsequent jobs from that source will have reverse protection disabled.	X	
Chained	You cannot protect a single source to a single target, where the target then acts as a source, sending the same data from the original source to a final target server.		X
Single server	You cannot protect a single source to itself.		X
Standalone to standalone	Your servers can be in a standalone to standalone configuration.	X	
Standalone to cluster	Your servers cannot be in a standalone to cluster configuration.		X
Cluster to standalone	Your servers cannot be in a cluster to standalone configuration.		X
Cluster to cluster	Your servers cannot be in a cluster to cluster configuration.		X

Target compatibility

- **Operating system version**—The source and target must have the same distribution and major version. For example, you cannot have a Red Hat version 5.8 source failing over to a Red Hat version 6.4 target. The two servers do not have to have the same minor version. For example, you can failover Red Hat version 6.4 to Red Hat version 6.5.
- **Source and target preparation**—Make sure your source and target servers are prepared for mirroring, replication, and failover by following these guidelines.
 - Uninstall any applications or operating system features that are not needed from both your source and target. Ideally, your target should be as clean and simple a configuration as possible.
 - Install on the source any drivers that are required on the target after failover. For example, you need to install on the source any NIC drivers that will be required on the target after failover.
 - Resolve any maintenance updates on the source that may require the server to be rebooted before failover.
 - Do not failover if the target is waiting on a reboot after applying maintenance. If failover occurs before the required reboot, the target may not operate properly or it may not boot.
- **Architecture**—The source and the target must have the same architecture. For example, you cannot failover a 32-bit server to a 64-bit server.
- **Processors**—There are no limits on the number or speed of the processors, but the source and the target should have at least the same number of processors. If the target has fewer processors or slower speeds than the source, there will be performance impacts for the users after failover.
- **Memory**—The target memory should be within 25% (plus or minus) of the source. If the target has much less memory than the source, there will be performance impacts for the users after failover.
- **Network adapters**—You must map at least one NIC from the source to one NIC on the target. If you have NICs on the source that are not being used, it is best to disable them. If the source has more NICs than the target, some of the source NICs will not be mapped to the target. Therefore, the IP addresses associated with those NICs will not be available after failover. If there are more NICs on the target than the source, the additional NICs will still be available after failover and will retain their pre-failover network settings.
- **File system format**—The source and the target must have the file system format on each server. For example, if you have Ext3 on the source, you cannot have XFS on the target. In that case, the target must also be Ext3.
- **Volumes**—There are no limits to the number of volumes you can protect on the source, although you are bound by operating system limits.

For each non-system volume you are protecting on the source, the target must have a matching volume. For example, if you are protecting /data and /home on the source, the target must also have /data and /home. Additional target volumes are preserved and available after failover with all data still accessible, however you will be unable to reverse protection if the target has more volumes than the source.

The system volumes / and /boot do not have this matching volume limitation. If you have / and /boot on different volumes on the source, they can exist on a single volume on the target. If you

have / and /boot on the same volume on the source, they can exist on different volumes on the target.

- **Carbonite Availability version**—If you will be using the reverse feature with your full server job, your source and target must be running the same Carbonite Availability version.
- **Disk space**—The target must have enough space to store the data from the source. This amount of disk space will depend on the applications and data files you are protecting. The more data you are protecting, the more disk space you will need. The target must also have enough space to store, process, and apply the source's system state data. If you will be enabling reverse protection, the source must have enough space to store, process, and apply the target's system state data.

A copy of the source data and system state will be staged on the target in a /dtstaging location for each mount point. For example, / will be staged in /dtstaging and /boot will be staged in /boot/dtstaging. For reverse protection, the same staging structure is used. You can predict how much space you will need in the staging folders by the amount of used space on the source or target, respectively.

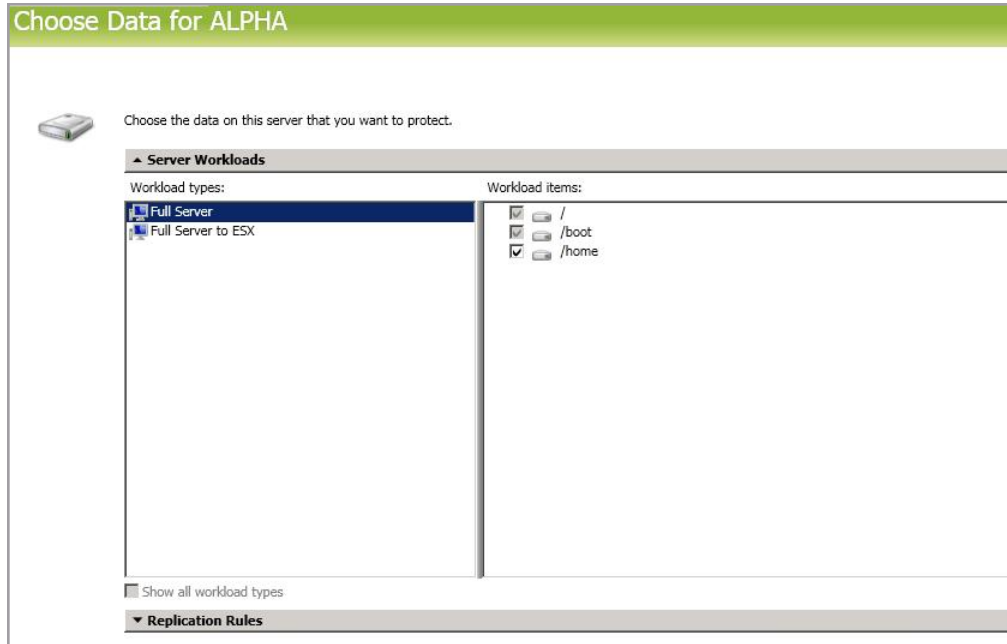
Keep in mind you should have extra space available on each server for any data growth.

- **Services**—Ideally, you should have the same services and run levels on the source and target.

Creating a full server job

Use these instructions to create a full server job.

1. From the **Servers** page, right-click the server you want to protect and select **Protect**. You can also highlight a server, click **Create a New Job** in the toolbar, then select **Protect**.
2. Choose the type of workload that you want to protect. Under **Server Workloads**, in the **Workload types** pane, select **Full Server**. In the **Workload items** pane, select the volumes on the source that you want to protect.



Unsupported file systems will be displayed but will not be accessible.

3. By default, Carbonite Availability selects the system and boot volumes for protection. You will be unable to deselect these volumes. Select any other volumes on the source that you want to protect.

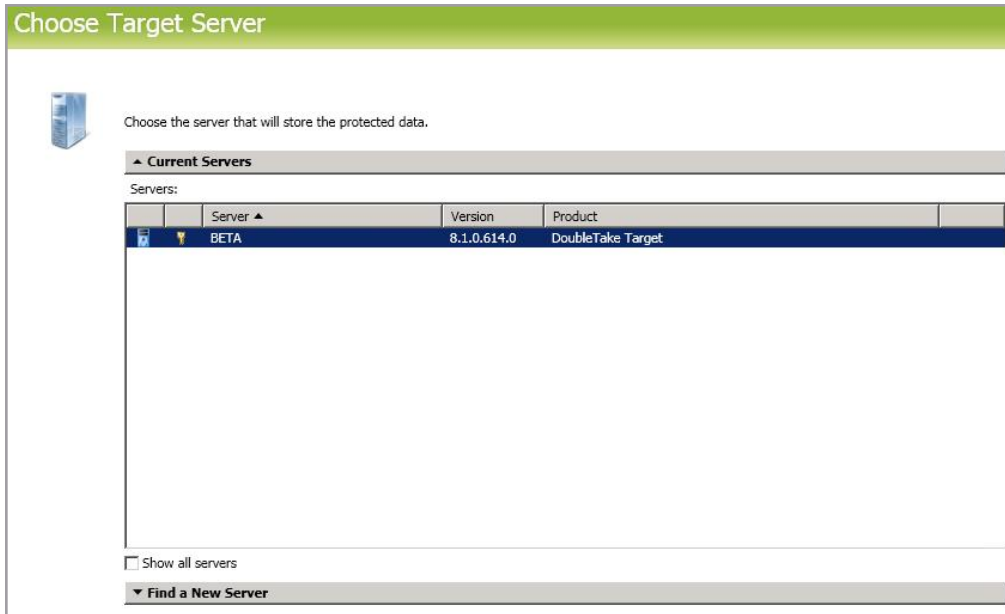
If desired, click the **Replication Rules** heading and expand the volumes under **Folders**. You will see that Carbonite Availability automatically excludes particular files that cannot be used during the protection. If desired, you can exclude other files that you do not want to protect, but be careful when excluding data. Excluded volumes, folders, and/or files may compromise the integrity of your installed applications.



If you return to this page using the **Back** button in the job creation workflow, your **Workload Types** selection will be rebuilt, potentially overwriting any manual replication rules that you specified. If you do return to this page, confirm your **Workload Types** and **Replication Rules** are set to your desired settings before proceeding forward again.

4. Click **Next** to continue.

5. Choose your target server. This is the server that will store the replica data from the source, and in the event of a failover, it will become your source.



- **Current Servers**—This list contains the servers currently available in your console session. Servers that are not licensed for the workflow you have selected and those not applicable to the workload type you have selected will be filtered out of the list. Select your target server from the list. If the server you are looking for is not displayed, enable **Show all servers**. The servers in red are not available for the source server or workload type you have selected. Hover your mouse over an unavailable server to see a reason why this server is unavailable.
- **Find a New Server**—If the server you need is not in the **Current Servers** list, click the **Find a New Server** heading. From here, you can specify a server along with credentials for logging in to the server. If necessary, you can click **Browse** to select a server from a network drill-down list.



If you enter the target server's fully-qualified domain name, the Carbonite Replication Console will resolve the entry to the server short name. If that short name resides in two different domains, this could result in name resolution issues. In this case, enter the IP address of the server.

When specifying credentials for a new server, specify a user that is a member of the local dtadmin security group.

6. Click **Next** to continue.



You may be prompted for a route from the target to the source. This route, and a port if you are using a non-default port, is used so the target can communicate with the source to build job options. This dialog box will be displayed, only if needed.

7. You have many options available for your Linux full server job. Configure those options that are applicable to your environment.

Go to each page identified below to see the options available for that section of the **Set Options** page. After you have configured your options, continue with the next step on page 200.

- *General* on page 187
- *Failover Monitor* on page 188
- *Test Failover* on page 190
- *Failover Options* on page 192
- *Failover Identity* on page 193
- *Reverse Protection and Routing* on page 194
- *Network Adapter Options* on page 196
- *Mirror, Verify & Orphaned Files* on page 197
- *Snapshots* on page 198
- *Compression* on page 199
- *Bandwidth* on page 200

General



The screenshot shows a window titled "General" with a small upward-pointing arrow icon to the left of the title. Below the title bar, there is a label "Job name:" followed by a text input field containing the text "alpha to beta".

For the **Job name**, specify a unique name for your job.

Failover Monitor

Failover Monitor

Total time to failure: 00:05:00

Consecutive failures: 20

Monitor on this interval: 00:00:10

Network monitoring

Monitor these addresses:

	Source IP Address
<input checked="" type="checkbox"/>	172.31.206.201

Monitoring method: Network service

Failover trigger: All monitored IP addresses fail

- **Total time to failure**—Specify, in hours:minutes:seconds, how long the target will keep trying to contact the source before the source is considered failed. This time is precise. If the total time has expired without a successful response from the source, this will be considered a failure.

Consider a shorter amount of time for servers, such as a web server or order processing database, which must remain available and responsive at all times. Shorter times should be used where redundant interfaces and high-speed, reliable network links are available to prevent the false detection of failure. If the hardware does not support reliable communications, shorter times can lead to premature failover. Consider a longer amount of time for machines on slower networks or on a server that is not transaction critical. For example, failover would not be necessary in the case of a server restart.

- **Consecutive failures**—Specify how many attempts the target will make to contact the source before the source is considered failed. For example, if you have this option set to 20, and your source fails to respond to the target 20 times in a row, this will be considered a failure.
- **Monitor on this interval**—Specify, in hours:minutes:seconds, how long to wait between attempts to contact the source to confirm it is online. This means that after a response (success or failure) is received from the source, Carbonite Availability will wait the specified interval time before contacting the source again. If you set the interval to 00:00:00, then a new check will be initiated immediately after the response is received.

If you choose **Total time to failure**, do not specify a longer interval than failure time or your server will be considered failed during the interval period.

- **Network monitoring**—With this option, the target will monitor the source using a network ping.
 - **Monitor these addresses**—Select each **Source IP Address** that you want the target to monitor. If you are using a NAT environment, do not select a private IP address on the source because the target cannot reach the source's private address,

thus causing an immediate failure. Also for NAT environments, you will see another column for the **Replication Service port**. This gives you the ability to specify the port number to be used with the address, allowing the target to monitor the source through a router.

- **Monitoring method**—This option determines the type of network ping used for failover monitoring.
 - **Network service**—Source availability will be tested by an ICMP ping to confirm the route is active.
 - **Replication service**—Source availability will be tested by a UDP ping to confirm the Double-Take service is active.
 - **Network and replication services**—Source availability will be tested by both an ICMP ping to confirm the route is active and a UDP ping to confirm the Double-Take service is active. If either monitoring method fails, failover will be triggered.
- **Failover trigger**—If you are monitoring multiple IP addresses, specify when you want a failover condition to be triggered.
 - **One monitored IP address fails**—A failover condition will be triggered when any one of the monitored IP addresses fails. If each IP address is on a different subnet, you may want to trigger failover after one fails.
 - **All monitored IP addresses fail**—A failover condition will be triggered when all monitored IP addresses fail. If there are multiple, redundant paths to a server, losing one probably means an isolated network problem and you should wait for all IP addresses to fail.

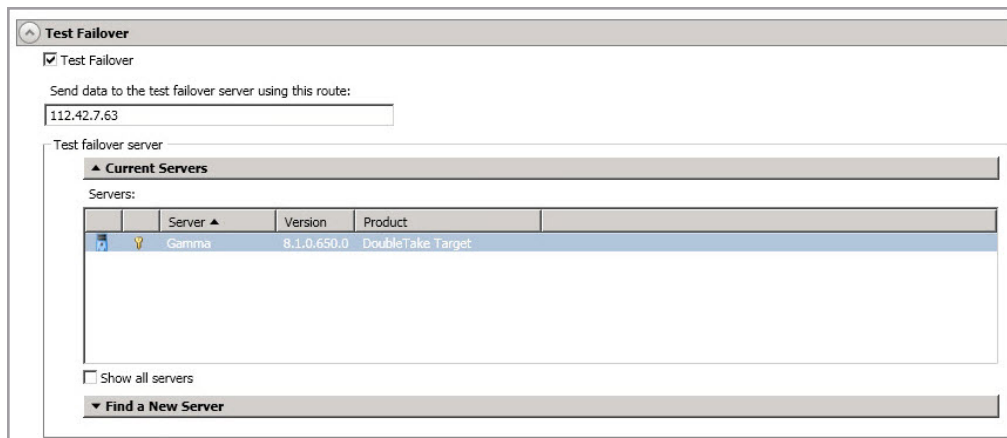
Test Failover

These options allow you to perform a test failover. Keep in mind the following for using test failover.



Test failover is not supported for Btrfs file systems.

- The source, target, and protection job will remain online and uninterrupted during the test.
- During the test, any scheduled snapshots for the protection job will be deferred until after the test. Manual snapshots will be disabled until after the test.
- The test will be performed using the test failover settings configured during job creation.
- The test failover will take a snapshot of the current data on the target and mirror the data from the snapshot to the test failover machine using the same mirroring options as the protection job.
- Once the mirror is complete, the test failover machine is rebooted automatically to finalize the test failover process.
- The test failover machine will maintain its own identity during the test.
- The test failover machine should maintain its own networking and be isolated from the rest of the network in order to avoid network conflicts and redirecting clients.
- When you are finished with your test, undo it.
- When you undo a test failover, the snapshot will be deleted.
- At any time during a test failover, you can undo the test, perform a live failover, or failover to a snapshot. (Performing a live failover or failing over to a snapshot will automatically undo any test in progress.)



- **Test Failover**—Enable this option to be able to perform test failover.
- **Send data to the test failover server using this route**—Enter a route to use on the test failover server for mirroring the data from the snapshot to the test failover server.
- **Test failover server**—Select the server you want to use for the test failover.
 - **Current Servers**—This list contains the servers currently available in your console session. Servers that are not applicable to test failover will be filtered out of the list. Select your test failover server from the list. If the server you are looking for is not

displayed, enable **Show all servers**. The servers in red are not available. Hover your mouse over an unavailable server to see a reason why this server is unavailable.

- **Find a New Server**—If the server you need is not in the **Current Servers** list, click the **Find a New Server** heading. From here, you can specify a server along with credentials for logging in to the server. If necessary, you can click **Browse** to select a server from a network drill-down list. After you have identified or located the server, click **Add Server**. If there are any issues connecting to that server, you will see an error in yellow at the top of the page. If there are no issues, you can continue.

Failover Options

The screenshot shows a window titled "Failover Options" with a close button. It contains a checked checkbox "Wait for user to initiate failover". Below this is a section titled "Target scripts" which is expanded. It contains two rows of configuration. The first row is for the "Pre-failover script" and includes a text input field, a browse button "...", and an "Arguments:" label followed by another text input field. The second row is for the "Post-failover script" and includes a text input field, a browse button "...", and an "Arguments:" label followed by another text input field. There is also an unchecked checkbox "Delay failover until script completes" located between the two script rows.

- **Wait for user to initiate failover**—The failover process can wait for you to initiate it, allowing you to control when failover occurs. When a failure occurs, the job will wait in **Failover Condition Met** for you to manually initiate the failover process. Disable this option if you want failover to occur immediately when a failure occurs.
- **Target Scripts**—You can customize failover by running scripts on the target. Scripts may contain any valid Linux command, executable, or shell script file. The scripts are processed using the same account running the Double-Take Management service. Examples of functions specified in scripts include stopping services on the target before failover because they may not be necessary, stopping services on the target that need to be restarted with the source's machine name and/or IP address, starting services or loading applications that are in an idle, standby mode waiting for failover to occur, notifying the administrator before and after failover occurs, and so on. There are two types of failover scripts.
 - **Pre-failover script**—This script runs on the target at the beginning of the failover process. Specify the full path and name of the script file.
 - **Delay until script completes**—Enable this option if you want to delay the failover process until the associated script has completed. If you select this option, make sure your script handles errors, otherwise the failover process may never complete if the process is waiting on a script that cannot complete.
 - **Post-failover script**—This script runs on the recovered source at the end of the failover process. Specify the full path and name of the script file.
 - **Arguments**—Specify a comma-separated list of valid arguments required to execute the script.

Failover Identity

Failover Identity

- Apply source network configuration to the target (Recommended for LAN configurations)
- Retain target network configuration (Recommended for WAN configurations)

- **Apply source network configuration to the target**—If you select this option, your source IP addresses will failover to the target. If your target is on the same subnet as the source (typical of a LAN environment), you should select this option. Do not select this option if you are using a NAT environment that has a different subnet on the other side of the router.



Do not apply the source network configuration to the target in a WAN environment unless you have a VPN infrastructure so that the source and target can be on the same subnet, in which case IP address failover will work the same as a LAN configuration. If you do not have a VPN, you will have to reconfigure the routers by moving the source's subnet from the source's physical network to the target's physical network. There are a number of issues to consider when designing a solution that requires router configuration to achieve IP address failover. Since the route to the source's subnet will be changed at failover, the source server must be the only system on that subnet, which in turn requires all server communications to pass through a router. Additionally, it may take several minutes or even hours for routing tables on other routers throughout the network to converge.

-
- **Retain target network configuration**—If you select this option, the target will retain all of its original IP addresses. If your target is on a different subnet (typical of a WAN or NAT environment), you should select this option.

Reverse Protection and Routing

Reverse Protection and Routing

Send data to the target server using this route:
172.31.206.207

Enable reverse protection

A reserved IP address permanently identifies each server so that failover and reverse can both be performed. The reserved IP addresses will not be moved on failover or reverse. These addresses will also be used to route the data in non-NAT environments.

Select a reserved IP address on the source:
10.10.10.29

Select a reserved IP address on the target:
10.10.10.30

- **Send data to the target server using this route**—By default, Carbonite Availability will select a target route for transmissions. If desired, specify an alternate route on the target that the data will be transmitted through. This allows you to select a different route for Carbonite Availability traffic. For example, you can separate regular network traffic and Carbonite Availability traffic on a machine with multiple IP addresses. You can also select or manually enter a public IP address (which is the public IP address of the server's router) if you are using a NAT environment. If you enter a public IP addresses, you will see additional fields allowing you to disable the default communication ports and specify other port numbers to use, allowing the target to communicate through a router. The **Management Service port** can be disregarded. It is used for other job types. The **Replication Service port** is used for data transmission.
- **Enable reverse protection**—After failover, your target server is lost. Reverse protection allows you to store a copy of the target's system state on the source server, so that the target server will not be lost. The reverse process will bring the target identity back on the source hardware and establish protection. After the reverse, the source (running on the original target hardware) will be protected to the target (running on the original source hardware).

If you do not use reverse protection, after a failover, your target server will be lost. In order to continue protecting your data, you will have to manually rebuild your original source and restart protection, which can be a long and complicated process. Also, if you disable reverse, you will lose the activated target license after failover.

You may want to consider having two IP addresses on each server. This will allow you to monitor and failover one (or more) IP addresses, while still leaving an IP address that does not get failed over. This IP address that is not failed over is called a reserved IP address and can be used for the reverse process. The reserved IP address remains with the server hardware. Ideally, the reserved IP address should not be used for production communications. The reserved IP address can be on the same or a different subnet from your production IP addresses, however if the subnet is different, it should be on a different network adapter. The reserved IP addresses will also be used to route Carbonite Availability data.

You do not have to have a second IP address on each server. It is acceptable to use the production IP address for reverse protection, as long as you are selecting the option to retain the target configuration.

- **Select a reserved IP address on the source**—Specify an IP address on the source which will be used to permanently identify the source server. The IP address you specify will not be failed over to the target in the event of a failure. This allows you to reverse protection back to the source after a failover.
 - **Send data to source after reverse using this route**—This field will only be displayed if the console recognizes that your source address is a public NAT address. In that case, you can specify the route and disable the default communication port and specify another port number to use for data transmission.
 - **Select a reserved IP address on the target**—Specify an IP address on the target which will be used to permanently identify the target server. The IP address you specify will not be lost during failover. This allows you to reverse protection back to the source after a failover. In a non-NAT environment, this address will override the target route above and be used to route the data to the target server.
-

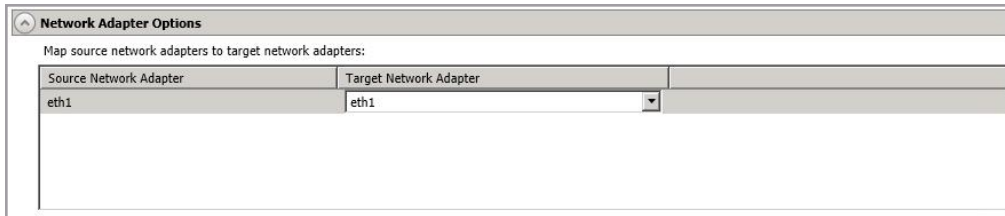


When reverse protection is enabled, your source server must have space to store, process, and apply the target's system state data.

When the job is first started and reverse protection is enabled, an image of the target's system state is mirrored to the source server. This mirror may cause a performance impact on your source server. This impact is only temporary, and system performance will return to normal when the reverse protection mirror is complete.

To maintain system performance on the source, the target's system state is not continuously replicated to the source. You can manually update the image of the target's system state by viewing the job details and clicking **Update** under **Target Server Image**. See *Viewing full server job details* on page 210.

Network Adapter Options



The screenshot shows a window titled "Network Adapter Options" with a sub-header "Map source network adapters to target network adapters:". Below this is a table with two columns: "Source Network Adapter" and "Target Network Adapter". The first row shows "eth1" in the source column and "eth1" in the target column. The target column has a dropdown arrow next to the text.

Source Network Adapter	Target Network Adapter
eth1	eth1

For **Map source network adapters to target network adapters**, specify how you want the IP addresses associated with each NIC on the source to be mapped to a NIC on the target. Do not mix public and private networks. Also, if you have enabled reverse protection, make sure that your NICs with your reserved IP addresses are mapped to each other.

Mirror, Verify & Orphaned Files

Mirror, Verify & Orphaned Files

Mirror Options

Choose a comparison method and whether to mirror the entire file or only the bytes that differ in each file.

Compare file attributes. Send the attributes and bytes that differ.

General Options

Delete orphaned files

This option cannot be changed for the current workload type

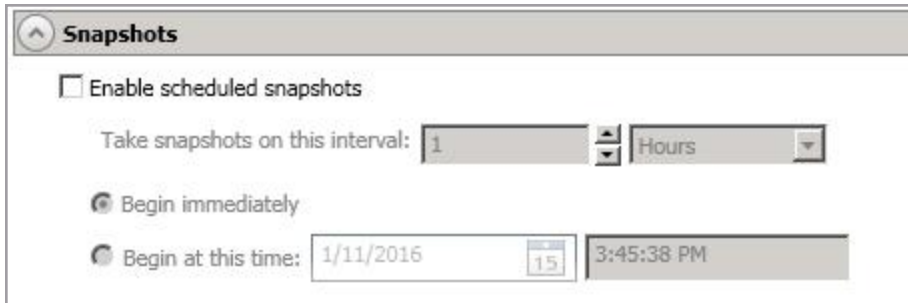
- **Mirror Options**—Choose a comparison method and whether to mirror the entire file or only the bytes that differ in each file.
 - **Do not compare files. Send the entire file.**—Carbonite Availability will not perform any comparisons between the files on the source and target. All files will be mirrored to the target, sending the entire file.
 - **Compare file attributes. Send the attributes and bytes that differ.**—Carbonite Availability will compare file attributes and will mirror only the attributes and bytes that are different.
 - **Compare file attributes and data. Send the attributes and bytes that differ.**—Carbonite Availability will compare file attributes and the file data and will mirror only the attributes and bytes that are different.
- **General Options**—Choose your general mirroring options.
 - **Delete orphaned files**—An orphaned file is a file that exists in the replica data on the target, but does not exist in the protected data on the source. This option specifies if orphaned files should be deleted on the target.



Orphaned file configuration is a per target configuration. All jobs to the same target will have the same orphaned file configuration.

If delete orphaned files is enabled, carefully review any replication rules that use wildcard definitions. If you have specified wildcards to be excluded from protection, files matching those wildcards will also be excluded from orphaned file processing and will not be deleted from the target. However, if you have specified wildcards to be included in your protection, those files that fall outside the wildcard inclusion rule will be considered orphaned files and will be deleted from the target.

Snapshots



Snapshots are not supported on Btrfs file systems or on servers running Ubuntu 12.04.x. See snapshots in *Full server requirements* on page 177 for complete details on the snapshot requirements.

A snapshot is an image of the source replica data on the target taken at a single point in time. You can failover to a snapshot. However, you cannot access the snapshot to recover specific files or folders.

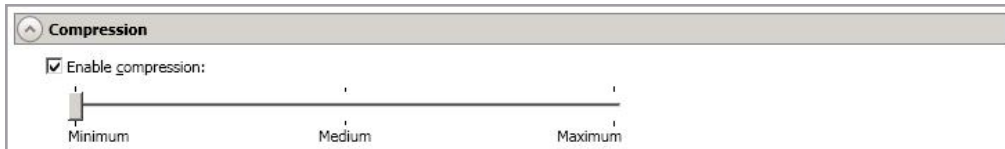
Turn on **Enable scheduled snapshots** if you want Carbonite Availability to take snapshots automatically at set intervals.

- **Take snapshots on this interval**—Specify the interval (in days, hours, or minutes) for taking snapshots.
- **Begin immediately**—Select this option if you want to start taking snapshots immediately after the protection job is established.
- **Begin at this time**—Select this option if you want to start taking snapshots at a later date and time. Specify the date and time parameters to indicate when you want to start.



See *Managing snapshots* on page 78 for details on taking manual snapshots and deleting snapshots.

Compression



To help reduce the amount of bandwidth needed to transmit Carbonite Availability data, compression allows you to compress data prior to transmitting it across the network. In a WAN environment this provides optimal use of your network resources. If compression is enabled, the data is compressed before it is transmitted from the source. When the target receives the compressed data, it decompresses it and then writes it to disk. You can set the level from **Minimum** to **Maximum** to suit your needs.

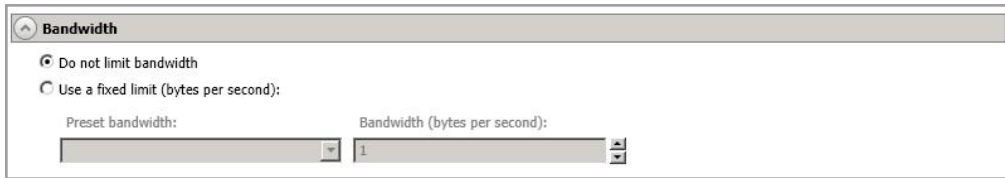
Keep in mind that the process of compressing data impacts processor usage on the source. If you notice an impact on performance while compression is enabled in your environment, either adjust to a lower level of compression, or leave compression disabled. Use the following guidelines to determine whether you should enable compression.

- If data is being queued on the source at any time, consider enabling compression.
- If the server CPU utilization is averaging over 85%, be cautious about enabling compression.
- The higher the level of compression, the higher the CPU utilization will be.
- Do not enable compression if most of the data is inherently compressed. Many image (.jpg, .gif) and media (.wmv, .mp3, .mpg) files, for example, are already compressed. Some images files, such as .bmp and .tif, are decompressed, so enabling compression would be beneficial for those types.
- Compression may improve performance even in high-bandwidth environments.
- Do not enable compression in conjunction with a WAN Accelerator. Use one or the other to compress Carbonite Availability data.



All jobs from a single source connected to the same IP address on a target will share the same compression configuration.

Bandwidth



Bandwidth limitations are available to restrict the amount of network bandwidth used for Carbonite Availability data transmissions. When a bandwidth limit is specified, Carbonite Availability never exceeds that allotted amount. The bandwidth not in use by Carbonite Availability is available for all other network traffic.



All jobs from a single source connected to the same IP address on a target will share the same bandwidth configuration.

- **Do not limit bandwidth**—Carbonite Availability will transmit data using 100% bandwidth availability.
 - **Use a fixed limit**—Carbonite Availability will transmit data using a limited, fixed bandwidth. Select a **Preset bandwidth** limit rate from the common bandwidth limit values. The **Bandwidth** field will automatically update to the bytes per second value for your selected bandwidth. This is the maximum amount of data that will be transmitted per second. If desired, modify the bandwidth using a bytes per second value. The minimum limit should be 3500 bytes per second.
8. Click **Next** to continue.
 9. Carbonite Availability validates that your source and target are compatible. The **Summary** page displays your options and validation items.

Errors are designated by a white X inside a red circle. Warnings are designated by a black exclamation point (!) inside a yellow triangle. A successful validation is designated by a white checkmark inside a green circle. You can sort the list by the icon to see errors, warnings, or successful validations together. Click on any of the validation items to see details. You must correct any errors before you can continue. Depending on the error, you may be able to click **Fix** or **Fix All** and let Carbonite Availability correct the problem for you. For those errors that Carbonite Availability cannot correct automatically, you will need to modify the source or target to correct the error, or you can select a different target. You must revalidate the selected servers, by clicking **Recheck**, until the validation check passes without errors.

If you receive a path transformation error during job validation indicating a volume does not exist on the target server, even though there is no corresponding data being protected on the source, you will need to manually modify your replication rules. Go back to the **Choose Data** page and under the **Replication Rules**, locate the volume from the error message. Remove any rules associated with that volume. Complete the rest of the workflow and the validation should pass.

After a job is created, the results of the validation checks are logged to the job log. See the *Carbonite Availability and Carbonite Move Reference Guide* for details on the various Carbonite Availability log files.

10. Once your servers have passed validation and you are ready to establish protection, click **Finish**,

and you will automatically be taken to the **Jobs** page.



Jobs in a NAT environment may take longer to start.

Once a job is created, do not change the name of underlying hardware components used in the job. For example, volume and datastore names or network adapter and virtual switch names. Any component used by name in your job must continue to use that name throughout the lifetime of job. If you must change a name, you will need to delete the job and re-create it using the new component name.

Managing and controlling full server jobs

Click **Jobs** from the main Carbonite Replication Console toolbar. The **Jobs** page allows you to view status information about your jobs. You can also control your jobs from this page.

The jobs displayed in the right pane depend on the server group folder selected in the left pane. Every job for each server in your console session is displayed when the **Jobs on All Servers** group is selected. If you have created and populated server groups (see *Managing servers* on page 55), then only the jobs associated with the server or target servers in that server group will be displayed in the right pane.

- *Overview job information displayed in the top right pane* on page 202
- *Detailed job information displayed in the bottom right pane* on page 205
- *Job controls* on page 207


Overview job information displayed in the top right pane


The top pane displays high-level overview information about your jobs. You can sort the data within a column in ascending and descending order. You can also move the columns to the left or right of each other to create your desired column order. The list below shows the columns in their default left to right order.


If you are using server groups, you can filter the jobs displayed in the top right pane by expanding the **Server Groups** heading and selecting a server group.


Column 1 (Blank)

The first blank column indicates the state of the job.

 A green circle with a white checkmark indicates the job is in a healthy state. No action is required.

 A yellow triangle with a black exclamation point indicates the job is in a pending or warning state. This icon is also displayed on any server groups that you have created that contain a job in a pending or warning state. Carbonite Availability is working or waiting on a pending process or attempting to resolve the warning state.

 A red circle with a white X indicates the job is in an error state. This icon is also displayed on any server groups that you have created that contain a job in an error state. You will need to investigate and resolve the error.

 The job is in an unknown state.

Job

The name of the job

Source Server

The name of the source. This could be the name or IP address of your source.

Target Server

The name of the target. This could be the name or IP address of your target.

Job Type

Each job type has a unique job type name. This job is a Full Server job. For a complete list of all job type names, press F1 to view the Carbonite Replication Console online help.

Activity

There are many different **Activity** messages that keep you informed of the job activity. Most of the activity messages are informational and do not require any administrator interaction. If you see error messages, check the job details. Keep in mind that **Idle** indicates console to server activity is idle, not that your servers are idle.

Mirror Status

- **Calculating**—The amount of data to be mirrored is being calculated.
- **In Progress**—Data is currently being mirrored.
- **Waiting**—Mirroring is complete, but data is still being written to the target.
- **Idle**—Data is not being mirrored.
- **Paused**—Mirroring has been paused.
- **Stopped**—Mirroring has been stopped.
- **Removing Orphans**—Orphan files on the target are being removed or deleted depending on the configuration.
- **Verifying**—Data is being verified between the source and target.
- **Unknown**—The console cannot determine the status.

Replication Status

- **Replicating**—Data is being replicated to the target.
- **Ready**—There is no data to replicate.
- **Pending**—Replication is pending.
- **Stopped**—Replication has been stopped.
- **Out of Memory**—Replication memory has been exhausted.
- **Failed**—The Double-Take service is not receiving replication operations from the Carbonite Availability driver. Check the Event Viewer for driver related issues.
- **Unknown**—The console cannot determine the status.

Transmit Mode

- **Active**—Data is being transmitted to the target.
- **Paused**—Data transmission has been paused.
- **Scheduled**—Data transmission is waiting on schedule criteria.
- **Stopped**—Data is not being transmitted to the target.
- **Error**—There is a transmission error.
- **Unknown**—The console cannot determine the status.

Operating System

The job type operating system

Detailed job information displayed in the bottom right pane

The details displayed in the bottom pane provide additional information for the job highlighted in the top pane. You can expand or collapse the bottom pane by clicking on the **Job Highlights** heading.

Name

The name of the job

Target data state

- **OK**—The data on the target is in a good state.
- **Mirroring**—The target is in the middle of a mirror process. The data will not be in a good state until the mirror is complete.
- **Mirror Required**—The data on the target is not in a good state because a remirror is required. This may be caused by an incomplete or stopped mirror or an operation may have been dropped on the target.
- **Busy**—The source is low on memory causing a delay in getting the state of the data on the target.
- **Not Loaded**—Carbonite Availability target functionality is not loaded on the target server. This may be caused by a license key error.
- **Not Ready**—The Linux drivers have not yet completed loading on the target.
- **Unknown**—The console cannot determine the status.

Mirror remaining

The total number of mirror bytes that are remaining to be sent from the source to the target.

Mirror skipped

The total number of bytes that have been skipped when performing a difference. These bytes are skipped because the data is not different on the source and target.

Replication queue

The total number of replication bytes in the source queue

Disk queue

The amount of disk space being used to queue data on the source

Recovery point latency

The length of time replication is behind on the target compared to the source. This is the time period of replication data that would be lost if a failure were to occur at the current time. This value represents replication data only and does not include mirroring data. If you are mirroring and failover, the data on the target will be at least as far behind as the replication point latency. It could potentially be further behind depending on the circumstances of the mirror. If mirroring is idle and you failover, the data will only be as far behind as the replication point latency time.

Bytes sent

The total number of mirror and replication bytes that have been transmitted to the target

Bytes sent (compressed)

The total number of compressed mirror and replication bytes that have been transmitted to the target. If compression is disabled, this statistic will be the same as **Bytes sent**.

Connected since

The date and time indicating when the current job was started.

Recent activity

Displays the most recent activity for the selected job, along with an icon indicating the success or failure of the last initiated activity. Click the link to see a list of recent activities for the selected job. You can highlight an activity in the list to display additional details about the activity.

Additional information

Depending on the current state of your job, you may see additional information displayed to keep you informed about the progress and status of your job. If there is no additional information, you will see (None) displayed.

Job controls

You can control your job through the toolbar buttons available on the **Jobs** page. If you select multiple jobs, some of the controls will apply only to the first selected job, while others will apply to all of the selected jobs. For example, **View Job Details** will only show details for the first selected job, while **Stop** will stop protection for all of the selected jobs.

If you want to control just one job, you can also right click on that job and access the controls from the pop-up menu.

View Job Details

This button leaves the **Jobs** page and opens the **View Job Details** page.

Edit Job Properties

This button leaves the **Jobs** page and opens the **EditJob Properties** page.

Delete

Stops (if running) and deletes the selected jobs.

Provide Credentials

Changes the login credentials that the job (which is on the target machine) uses to authenticate to the servers in the job. This button opens the Provide Credentials dialog box where you can specify the new account information and which servers you want to update. See *Providing server credentials* on page 66. You will remain on the **Jobs** page after updating the server credentials. If your servers use the same credentials, make sure you also update the credentials on the **Servers** page so that the Carbonite Replication Console can authenticate to the servers in the console session. See *Managing servers* on page 55.

View Recent Activity

Displays the recent activity list for the selected job. Highlight an activity in the list to display additional details about the activity.

Start

Starts or resumes the selected jobs.

If you have previously stopped protection, the job will restart mirroring and replication.

If you have previously paused protection, the job will continue mirroring and replication from where it left off, as long as the Carbonite Availability queue was not exhausted

during the time the job was paused. If the Carbonite Availability queue was exhausted during the time the job was paused, the job will restart mirroring and replication.

Also if you have previously paused protection, all jobs from the same source to the same IP address on the target will be resumed.

Pause

Pauses the selected jobs. Data will be queued on the source while the job is paused.

All jobs from the same source to the same IP address on the target will be paused.

Stop

Stops the selected jobs. The jobs remain available in the console, but there will be no mirroring or replication data transmitted from the source to the target. Mirroring and replication data will not be queued on the source while the job is stopped, requiring a remirror when the job is restarted. The type of remirror will depend on your job settings.

Take Snapshot

Even if you have scheduled the snapshot process, you can run it manually any time. If an automatic or scheduled snapshot is currently in progress, Carbonite Availability will wait until that one is finished before taking the manual snapshot.

Snapshots are not supported on Btrfs file systems or on servers running Ubuntu 12.04.x. See snapshots in *Full server requirements* on page 177 for complete details on the snapshot requirements.

Manage Snapshots

Allows you to manage your snapshots by taking and deleting snapshots for the selected job. See *Managing snapshots* on page 78 for more information.

Failover or Cutover

Starts the failover process. See *Failing over full server jobs* on page 219 for the process and details of failing over a Linux full server job.

Failback

Starts the failback process. Failback does not apply to full server for Linux jobs.

Restore

Starts the restoration process. Restoration does not apply to full server for Linux jobs.

Reverse

Reverses protection. The original source hardware will be reversed to the target identity and the job will start mirroring in the reverse direction with the job name and log file names changing accordingly. After the mirror is complete, the job will continue running in the opposite direction. See *Reversing full server jobs* on page 221 for the process and details of reversing a full server job.

Undo Failover or Cutover

Cancels a test failover by undoing it. Undo failover does not apply to full server for Linux jobs.

View Job Log

Opens the job log. On the right-click menu, this option is called **View Logs**, and you have the option of opening the job log, source server log, or target server log.

Other Job Actions

Opens a small menu of other job actions. These job actions are not available for Linux jobs.

Filter

Select a filter option from the drop-down list to only display certain jobs. You can display **Healthy jobs**, **Jobs with warnings**, or **Jobs with errors**. To clear the filter, select **All jobs**. If you have created and populated server groups, then the filter will only apply to the jobs associated with the server or target servers in that server group. See *Managing servers* on page 55.

Search

Allows you to search the source or target server name for items in the list that match the criteria you have entered.

Overflow Chevron

Displays any toolbar buttons that are hidden from view when the window size is reduced.

Viewing full server job details

From the **Jobs** page, highlight the job and click **View Job Details** in the toolbar.

Review the following table to understand the detailed information about your job displayed on the **View Job Details** page.





Job name

The name of the job

Job type

Each job type has a unique job type name. This job is a Full Server job. For a complete list of all job type names, press F1 to view the Carbonite Replication Console online help.

Health

-  The job is in a healthy state.
-  The job is in a warning state.
-  The job is in an error state.
-  The job is in an unknown state.

Activity

There are many different **Activity** messages that keep you informed of the job activity. Most of the activity messages are informational and do not require any administrator interaction. If you see error messages, check the rest of the job details.

Connection ID

The incremental counter used to number connections. The number is incremented when a connection is created. The counter is reset if there are no existing jobs and the Double-Take service is restarted.

Transmit mode

- **Active**—Data is being transmitted to the target.
- **Paused**—Data transmission has been paused.
- **Scheduled**—Data transmission is waiting on schedule criteria.
- **Stopped**—Data is not being transmitted to the target.
- **Error**—There is a transmission error.
- **Unknown**—The console cannot determine the status.

Target data state

- **OK**—The data on the target is in a good state.
- **Mirroring**—The target is in the middle of a mirror process. The data will not be in a good state until the mirror is complete.
- **Mirror Required**—The data on the target is not in a good state because a remirror is required. This may be caused by an incomplete or stopped mirror or an operation may have been dropped on the target.
- **Busy**—The source is low on memory causing a delay in getting the state of the data on the target.
- **Not Loaded**—Carbonite Availability target functionality is not loaded on the target server. This may be caused by a license key error.
- **Not Ready**—The Linux drivers have not yet completed loading on the target.
- **Unknown**—The console cannot determine the status.

Target route

The IP address on the target used for Carbonite Availability transmissions.

Compression

- **On / Level**—Data is compressed at the level specified.
- **Off**—Data is not compressed.

Encryption

- **On**—Data is being encrypted before it is sent from the source to the target.
- **Off**—Data is not being encrypted before it is sent from the source to the target.

Bandwidth limit

If bandwidth limiting has been set, this statistic identifies the limit. The keyword **Unlimited** means there is no bandwidth limit set for the job.

Connected since

The source server date and time indicating when the current job was started. This field is blank, indicating that a TCP/IP socket is not present, when the job is waiting on transmit options or if the transmission has been stopped. This field will maintain the date and time, indicating that a TCP/IP socket is present, when transmission has been paused.

Additional information

Depending on the current state of your job, you may see additional information displayed to keep you informed about the progress and status of your job. If there is no additional information, you will see (None) displayed.

Mirror status

- **Calculating**—The amount of data to be mirrored is being calculated.
- **In Progress**—Data is currently being mirrored.
- **Waiting**—Mirroring is complete, but data is still being written to the target.

- **Idle**—Data is not being mirrored.
- **Paused**—Mirroring has been paused.
- **Stopped**—Mirroring has been stopped.
- **Removing Orphans**—Orphan files on the target are being removed or deleted depending on the configuration.
- **Verifying**—Data is being verified between the source and target.
- **Unknown**—The console cannot determine the status.

Mirror percent complete

The percentage of the mirror that has been completed

Mirror remaining

The total number of mirror bytes that are remaining to be sent from the source to the target.

Mirror skipped

The total number of bytes that have been skipped when performing a difference. These bytes are skipped because the data is not different on the source and target.

Replication status

- **Replicating**—Data is being replicated to the target.
- **Ready**—There is no data to replicate.
- **Pending**—Replication is pending.
- **Stopped**—Replication has been stopped.
- **Out of Memory**—Replication memory has been exhausted.
- **Failed**—The Double-Take service is not receiving replication operations from the Carbonite Availability driver. Check the Event Viewer for driver related issues.
- **Unknown**—The console cannot determine the status.

Replication queue

The total number of replication bytes in the source queue

Disk queue

The amount of disk space being used to queue data on the source

Bytes sent

The total number of mirror and replication bytes that have been transmitted to the target

Bytes sent compressed

The total number of compressed mirror and replication bytes that have been transmitted to the target. If compression is disabled, this statistic will be the same as **Bytes sent**.

Recovery point latency

The length of time replication is behind on the target compared to the source. This is the time period of replication data that would be lost if a failure were to occur at the current time. This value represents replication data only and does not include mirroring data. If you are mirroring and failover, the data on the target will be at least as far behind as the replication point latency. It could potentially be further behind depending on the circumstances of the mirror. If mirroring is idle and you failover, the data will only be as far behind as the replication point latency time.

Mirror start time

The UTC time when mirroring started

Mirror end time

The UTC time when mirroring ended

Total time for last mirror

The length of time it took to complete the last mirror process

Target Server Image

When a full server job is created with reverse protection enabled, an image of the target's system state is stored on the source server. This image allows you to reverse your source and target after a failover. To improve performance, the target's system state is not continuously replicated to the source. You should manually update the image of the target's system state by clicking **Update** if there is a change on the target. For example, if the credentials on the target server are updated, you should update the target server image that is on the source. This reverse protection mirror may cause a performance impact on your source server. This impact is only temporary, and system performance will return to normal when the reverse protection mirror is complete.

If you have reverse enabled, are updating your target image, and the Double-Take service on the target restarts (either manually or automatically, for example the target server restarts), you should restart your target image update after the Double-Take service is back online. This will correct any incorrect status displayed in the console and ensure the target image is complete.

Validating a full server job

Over time, you may want to confirm that any changes in your network or environment have not impacted your Carbonite Availability job. Use these instructions to validate an existing job.

1. From the **Jobs** page, highlight the job and click **View Job Details** in the toolbar.
2. In the **Tasks** area on the right on the **View Job Details** page, click **Validate job properties**.
3. Errors are designated by a white X inside a red circle. Warnings are designated by a black exclamation point (!) inside a yellow triangle. A successful validation is designated by a white checkmark inside a green circle. You can sort the list by the icon to see errors, warnings, or successful validations together. Click on any of the validation items to see details. You must correct any errors before you can continue. Depending on the error, you may be able to click **Fix** or **Fix All** and let Carbonite Availability correct the problem for you. For those errors that Carbonite Availability cannot correct automatically, you will need to modify the source or target to correct the error, or you can select a different target. You must revalidate the selected servers, by clicking **Recheck**, until the validation check passes without errors.
4. Once your servers have passed validation, click **Close**.

Editing a full server job

Use these instructions to edit a full server job.

1. From the **Jobs** page, highlight the job and click **View Job Details** in the toolbar.
2. In the **Tasks** area on the right on the **View Job Details** page, click **Edit job properties**.
3. You will see the same options for your full server job as when you created the job, but you will not be able to edit all of them. If desired, edit those options that are configurable for an existing job. See *Creating a full server job* on page 184 for details on each job option.



Changing some options may require Carbonite Availability to automatically disconnect, reconnect, and remirror the job.

-
4. If you want to modify the workload items or replication rules for the job, click **Edit workload or replication rules**. Modify the **Workload item** you are protecting, if desired. Additionally, you can modify the specific **Replication Rules** for your job.

Volumes and folders with a green highlight are included completely. Volumes and folders highlighted in light yellow are included partially, with individual files or folders included. If there is no highlight, no part of the volume or folder is included. To modify the items selected, highlight a volume, folder, or file and click **Add Rule**. Specify if you want to **Include** or **Exclude** the item. Also, specify if you want the rule to be recursive, which indicates the rule should automatically be applied to the subdirectories of the specified path. If you do not select **Recursive**, the rule will not be applied to subdirectories.

Click **OK** to return to the **Edit Job Properties** page.



If you remove data from your workload and that data has already been sent to the target, you will need to manually remove that data from the target. Because the data you removed is no longer included in the replication rules, Carbonite Availability orphan file detection cannot remove the data for you. Therefore, you have to remove it manually.

-
5. Click **Next** to continue.
 6. Carbonite Availability validates that your source and target are compatible. The **Summary** page displays your options and validation items.

Errors are designated by a white X inside a red circle. Warnings are designated by a black exclamation point (!) inside a yellow triangle. A successful validation is designated by a white checkmark inside a green circle. You can sort the list by the icon to see errors, warnings, or successful validations together. Click on any of the validation items to see details. You must correct any errors before you can continue. Depending on the error, you may be able to click **Fix** or **Fix All** and let Carbonite Availability correct the problem for you. For those errors that Carbonite Availability cannot correct automatically, you will need to modify the source or target to correct the error, or you can select a different target. You must revalidate the selected servers, by clicking **Recheck**, until the validation check passes without errors.

If you receive a path transformation error during job validation indicating a volume does not exist on the target server, even though there is no corresponding data being protected on the source,

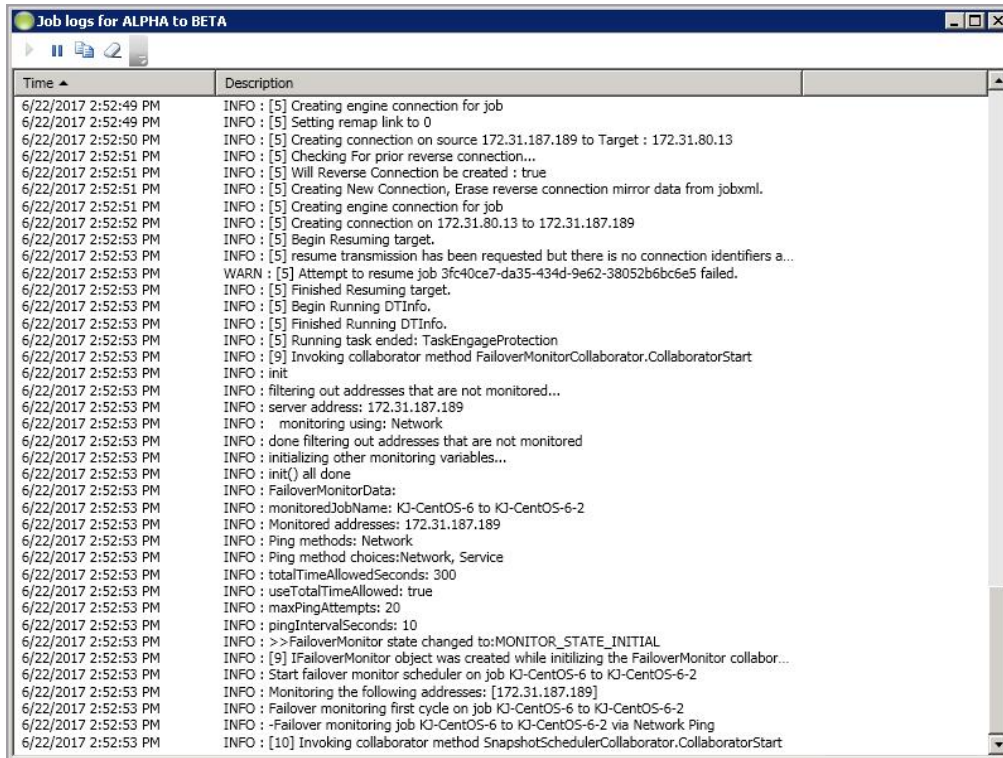
you will need to manually modify your replication rules. Go back to the **Choose Data** page and under the **Replication Rules**, locate the volume from the error message. Remove any rules associated with that volume. Complete the rest of the workflow and the validation should pass.

After a job is created, the results of the validation checks are logged to the job log. See the *Carbonite Availability and Carbonite Move Reference Guide* for details on the various Carbonite Availability log files.

7. Once your servers have passed validation and you are ready to update your job, click **Finish**.

Viewing a full server job log

You can view a job log file through the Carbonite Replication Console by selecting **View Job Log** from the toolbar on the **Jobs** page. Separate logging windows allow you to continue working in the Carbonite Replication Console while monitoring log messages. You can open multiple logging windows for multiple jobs. When the Carbonite Replication Console is closed, all logging windows will automatically close.



Time	Description
6/22/2017 2:52:49 PM	INFO : [5] Creating engine connection for job
6/22/2017 2:52:49 PM	INFO : [5] Setting remap link to 0
6/22/2017 2:52:50 PM	INFO : [5] Creating connection on source 172.31.187.189 to Target : 172.31.80.13
6/22/2017 2:52:51 PM	INFO : [5] Checking For prior reverse connection...
6/22/2017 2:52:51 PM	INFO : [5] Will Reverse Connection be created : true
6/22/2017 2:52:51 PM	INFO : [5] Creating New Connection, Erase reverse connection mirror data from jobxml.
6/22/2017 2:52:51 PM	INFO : [5] Creating engine connection for job
6/22/2017 2:52:52 PM	INFO : [5] Creating connection on 172.31.80.13 to 172.31.187.189
6/22/2017 2:52:53 PM	INFO : [5] Begin Resuming target.
6/22/2017 2:52:53 PM	INFO : [5] resume transmission has been requested but there is no connection identifiers a...
6/22/2017 2:52:53 PM	WARN : [5] Attempt to resume job 3fc40ce7-da35-434d-9e62-38052b6bc6e5 failed.
6/22/2017 2:52:53 PM	INFO : [5] Finished Resuming target.
6/22/2017 2:52:53 PM	INFO : [5] Begin Running DTInfo.
6/22/2017 2:52:53 PM	INFO : [5] Finished Running DTInfo.
6/22/2017 2:52:53 PM	INFO : [5] Running task ended: TaskEngageProtection
6/22/2017 2:52:53 PM	INFO : [9] Invoking collaborator method FailoverMonitorCollaborator.CollaboratorStart
6/22/2017 2:52:53 PM	INFO : init
6/22/2017 2:52:53 PM	INFO : filtering out addresses that are not monitored...
6/22/2017 2:52:53 PM	INFO : server address: 172.31.187.189
6/22/2017 2:52:53 PM	INFO : monitoring using: Network
6/22/2017 2:52:53 PM	INFO : done filtering out addresses that are not monitored
6/22/2017 2:52:53 PM	INFO : initializing other monitoring variables...
6/22/2017 2:52:53 PM	INFO : init() all done
6/22/2017 2:52:53 PM	INFO : FailoverMonitorData:
6/22/2017 2:52:53 PM	INFO : monitoredJobName: KJ-CentOS-6 to KJ-CentOS-6-2
6/22/2017 2:52:53 PM	INFO : Monitored addresses: 172.31.187.189
6/22/2017 2:52:53 PM	INFO : Ping methods: Network
6/22/2017 2:52:53 PM	INFO : Ping method choices:Network, Service
6/22/2017 2:52:53 PM	INFO : totalTimeAllowedSeconds: 300
6/22/2017 2:52:53 PM	INFO : useTotalTimeAllowed: true
6/22/2017 2:52:53 PM	INFO : maxPingAttempts: 20
6/22/2017 2:52:53 PM	INFO : pingIntervalSeconds: 10
6/22/2017 2:52:53 PM	INFO : >>FailoverMonitor state changed to:MONITOR_STATE_INITIAL
6/22/2017 2:52:53 PM	INFO : [9] IFailoverMonitor object was created while initializing the FailoverMonitor collabor...
6/22/2017 2:52:53 PM	INFO : Start failover monitor scheduler on job KJ-CentOS-6 to KJ-CentOS-6-2
6/22/2017 2:52:53 PM	INFO : Monitoring the following addresses: [172.31.187.189]
6/22/2017 2:52:53 PM	INFO : Failover monitoring first cycle on Job KJ-CentOS-6 to KJ-CentOS-6-2
6/22/2017 2:52:53 PM	INFO : -Failover monitoring job KJ-CentOS-6 to KJ-CentOS-6-2 via Network Ping
6/22/2017 2:52:53 PM	INFO : [10] Invoking collaborator method SnapshotSchedulerCollaborator.CollaboratorStart

The following table identifies the controls and the table columns in the **Job logs** window.

Start 

This button starts the addition and scrolling of new messages in the window.

Pause 

This button pauses the addition and scrolling of new messages in the window. This is only for the **Job logs** window. The messages are still logged to their respective files on the server.

Copy 

This button copies the messages selected in the **Job logs** window to the Windows clipboard.

Clear

This button clears the **Job logs** window. The messages are not cleared from the respective files on the server. If you want to view all of the messages again, close and reopen the **Job logs** window.

Time

This column in the table indicates the date and time when the message was logged.

Description

This column in the table displays the actual message that was logged.

Failing over full server jobs

You will be notified in the console when a failover condition has been met. At this time, you should trigger failover. You can also trigger failover at any other time you desire, thus allowing you to better control the failover process.



Resolve any maintenance updates on the source that may require the server to be rebooted before failover or failback. Also, do not failover or failback if the target is waiting on a reboot after applying maintenance. If failover occurs before the required reboot, the target may not operate properly or it may not boot.

-
1. On the **Jobs** page, highlight the job that you want to failover and click **Failover, Cutover, or Recover** in the toolbar.
 2. Select the type of failover to perform.
 - **Failover to live data**—Select this option to initiate a full, live failover using the current data on the target. The source is automatically shut down if it is still running. Then the target will stand in for the source by rebooting and applying the source identity, including its system state, on the target. After the reboot, the target becomes the source, and the target no longer exists.
 - **Perform test failover**—Select this option to perform a test failover.



Test failover is not supported for Btrfs file systems.

-
- The source, target, and protection job will remain online and uninterrupted during the test.
 - During the test, any scheduled snapshots for the protection job will be deferred until after the test. Manual snapshots will be disabled until after the test.
 - The test will be performed using the test failover settings configured during job creation.
 - The test failover will take a snapshot of the current data on the target and mirror the data from the snapshot to the test failover machine using the same mirroring options as the protection job.
 - Once the mirror is complete, the test failover machine is rebooted automatically to finalize the test failover process.
 - The test failover machine will maintain its own identity during the test.
 - The test failover machine should maintain its own networking and be isolated from the rest of the network in order to avoid network conflicts and redirecting clients.
 - When you are finished with your test, undo it.
 - When you undo a test failover, the snapshot will be deleted.
 - At any time during a test failover, you can undo the test, perform a live failover, or failover to a snapshot. (Performing a live failover or failing over to a snapshot will automatically undo any test in progress.)

- **Failover to a snapshot**—Select this option to initiate a full, live failover without using the current data on the target. Instead, select a snapshot and the data on the target will be reverted to that snapshot. This option will not be available if there are no snapshots on the target. To help you understand what snapshots are available, the **Type** indicates the kind of snapshot.
 - **Scheduled**—This snapshot was taken as part of a periodic snapshot.
 - **Deferred**—This snapshot was taken as part of a periodic snapshot, although it did not occur at the specified interval because the job between the source and target was not in a good state.
 - **Manual**—This snapshot was taken manually by a user.



Snapshots are not supported on Btrfs file systems or on servers running Ubuntu 12.04.x. See snapshots in *Full server requirements* on page 177 for complete details on the snapshot requirements.

3. Select how you want to handle the data in the target queue.
 - **Apply data in target queues before failover or cutover**—All of the data in the target queue will be applied before failover begins. The advantage to this option is that all of the data that the target has received will be applied before failover begins. The disadvantage to this option is depending on the amount of data in queue, the amount of time to apply all of the data could be lengthy.
 - **Discard data in the target queues and failover or cutover immediately**—All of the data in the target queue will be discarded and failover will begin immediately. The advantage to this option is that failover will occur immediately. The disadvantage is that any data in the target queue will be lost.
4. When you are ready to begin failover, click **Failover**.
5. If you performed a test failover, you can undo it by selecting **Undo Failover or Cutover** in the toolbar. If configured, the snapshots used for the test failover will be deleted.



If you need to update DNS after failover, there is a sample DNS update script located in `/etc/DT/sysprep.d`. You may need to modify the script for your environment. If you need basic assistance with script modifications, contact technical support. Assistance with advanced scripting will be referred to Professional Services.

Reversing full server jobs

After a full server failover, the source is running on your original target hardware and your target no longer exists. That means the source and target hardware now share the same identity, which is the source identity.



If you did not enable reverse protection or if you have to rebuild your source, you will have to reverse your protection manually.

1. Fix the issue that caused your original source server to fail.
2. Connect the original source server to the network.
3. Make sure the production NIC on your original source is online. If the NIC is disabled or unplugged, you will not be able to reverse. Make sure you continue to access the servers through the reserved IP addresses, but you can disregard any IP address conflicts for the primary NIC. Since the new source (running on the original target hardware) already has the source's address assigned to it, the source reserved IP address (set during the job creation workflow) will be used to identify the source. The machine names for both servers will be the same at this point. The reserved IP addresses which were selected during the job creation will be shown in parenthesis to identify the machines.
4. On the **Jobs** page, highlight the job that you want to reverse. If the job is not listed, you may need to add your servers to your console again. Use the reserved IP addresses and local credentials.
5. Highlight the job you want to reverse and click **Reverse** in the toolbar. During the reverse process, you will see various states for the job. The **Reversing** state will be displayed when the target identity is being established on the original source hardware. When the reverse process is complete, the target (on the original source hardware) will reboot. At this point, your source is still running on your original target hardware with the source name, but the original source hardware now has the target identity. After reboot, the job will start synchronizing. During the synchronizing process, protection is being established from the source (on the original target hardware) to the target (on the original source hardware). The reverse protection is also established in the opposite direction.
6. To go back to your original hardware, highlight the job and click **Failover, Cutover, or Recover**. The source identity will now be applied to the target (on the original source hardware), and the target identity will again be gone. Both servers will have the source identity.
7. To bring back the target identity, highlight the job and click **Reverse**. The same process as above will be repeated, but on the opposite servers. When the reverse is complete, you will be back to your original identities on the original hardware.

Chapter 5 Full server to ESX protection

Create a full server to ESX job when you want to protect an entire physical server or virtual machine to an ESX target. There is no reverse protection for this job. You will have to use another full server job type to get back to your original hardware after failover.

For full server to ESX protection, you will need to complete the following steps, in order.

1. Review the *Full server to ESX requirements* on page 223 to make sure your environment meets the requirements.
2. Deploy your Carbonite Availability virtual recovery appliance. See the *Carbonite Availability and Carbonite Move Installation, Licensing, and Activation* document for details.
3. Install the Carbonite Replication Console on a Windows machine. See the *Carbonite Availability and Carbonite Move Installation, Licensing, and Activation* document for details.
4. Add your virtual recovery appliance to the Carbonite Replication Console. See *Adding servers* on page 64.
5. Install Carbonite Availability on your Linux source servers. See the *Carbonite Availability and Carbonite Move Installation, Licensing, and Activation* document for details.
6. Add your source servers to your Carbonite Replication Console. See *Adding servers* on page 64.
7. Create your full server to ESX appliance job. See *Creating a full server to ESX job* on page 229.

Once your job is created and running, see the following sections to manage your job.

- *Managing and controlling full server to ESX jobs* on page 255—You can view status information about your job and learn how to control the job.
- *Failing over full server to ESX jobs* on page 272—Use this section when a failover condition has been met or whenever you want to failover.
- *Reversing protection after failover for full server to ESX jobs* on page 274—Use this section if you need to get your data back to the original hardware.

Full server to ESX requirements

Use these requirements for full server to ESX protection.

- **Source server**—The source server can be a physical or virtual server running any of the following operating systems.
 - **Operating system**—Red Hat Enterprise Linux, Oracle Enterprise Linux, and CentOS
 - **Version**—5.9 through 5.11
 - **Kernel type for x86 (32-bit) architectures**—Default, SMP, Xen, PAE
 - **Kernel type for x86-64 (64-bit) architectures**—Default, SMP, Xen
 - **File system**—Ext3, Ext4, XFS
 - **Notes**—Oracle Enterprise Linux support is for the mainline kernel only, not the Unbreakable kernel.
 - **Operating system**—Red Hat Enterprise Linux, Oracle Enterprise Linux, and CentOS
 - **Version**—6.7 through 6.9
 - **Kernel type for x86 (32-bit) architectures**—Default
 - **Kernel type for x86-64 (64-bit) architectures**—Default
 - **File system**—Ext3, Ext4, XFS (64-bit only)
 - **Operating system**—Red Hat Enterprise Linux, Oracle Enterprise Linux, and CentOS
 - **Version**—7.2 through 7.4
 - **Kernel type for x86 (32-bit) architectures**—No 32-bit architectures are supported
 - **Kernel type for x86-64 (64-bit) architectures**—Default
 - **File system**—Ext3, Ext4, XFS
 - **Notes**—Oracle Enterprise Linux version 7.4 is for the Mainline kernel only.
 - **Operating system**—SUSE Linux Enterprise
 - **Version**—11.2 through 11.4
 - **Kernel type for x86 (32-bit) architectures**—Default, Xen, XenPAE, VMI
 - **Kernel type for x86-64 (64-bit) architectures**—Default, Xen
 - **File system**—Ext3, XFS
 - **Operating system**—SUSE Linux Enterprise
 - **Version**—12.0 through 12.2
 - **Kernel type for x86 (32-bit) architectures**—No 32-bit architectures are supported
 - **Kernel type for x86-64 (64-bit) architectures**—Default
 - **File system**—Ext3, Ext4, XFS, Btrfs
 - **Notes**—If you are planning to convert an existing file system to Btrfs, you must delete any existing Carbonite Availability jobs and re-create them after converting to Btrfs.
 - **Operating system**—Ubuntu
 - **Version**—12.04.3, 12.04.4, and 12.04.5
 - **Kernel type for x86 (32-bit) architectures**—Generic

- **Kernel type for x86-64 (64-bit) architectures**—Generic
- **File system**—Ext2, Ext3, Ext4, XFS
- **Operating system**—Ubuntu
 - **Version**—14.04.3, 14.04.4, and 14.04.5
 - **Kernel type for x86 (32-bit) architectures**—Generic
 - **Kernel type for x86-64 (64-bit) architectures**—Generic
 - **File system**—Ext2, Ext3, Ext4, XFS
- **Operating system**—Ubuntu
 - **Version**—16.04.0, 16.04.1, and 16.04.2
 - **Kernel type for x86 (32-bit) architectures**—Generic
 - **Kernel type for x86-64 (64-bit) architectures**—Generic
 - **File system**—Ext2, Ext3, Ext4, XFS



For all operating systems except Ubuntu, the kernel version must match the expected kernel for the specified release version. For example, if `/etc/redhat-release` declares the system to be a Redhat 6.6 system, the kernel that is installed must match that.

Carbonite Availability does not support stacking filesystems, like eCryptFS.

- **Packages and services**—Each Linux server must have the following packages and services installed before you can install and use Carbonite Availability. See your operating system documentation for details on these packages and utilities.
 - sshd (or the package that installs sshd)
 - parted
 - `/usr/bin/which`
 - `/usr/sbin/dmidecode`
 - `/usr/bin/scp` (only if you will be performing push installations from the Carbonite Replication Console to your Linux servers)
 - `/lib/lsb/init-functions` (only if you are installing on Red Hat, Oracle Enterprise Linux, CentOS, or SUSE)
- **vCenter**—vCenter is not required, but if you are using it, then you must use version 5.5 or later. If you upgrade your version of vCenter after it has been entered into the Carbonite Replication Console, you must remove and re-add the vCenter in order for the console to recognize the upgraded version.
- **vMotion**—Host vMotion is only supported if you are using vCenter. Storage vMotion is not supported.
- **Target host server**—The target host server must be an ESX server. It can be any of the following ESX operating systems.
 - ESXi 5.5
 - ESXi 6.0
 - ESXi 6.5



The free versions of ESX restrict functionality that Carbonite Availability requires. Therefore, you must use one of the paid editions of ESX.

- **Virtual recovery appliance**—The target ESX host must have an existing virtual machine, known as a virtual recovery appliance. You must have this appliance before you can begin protection. When you begin protection, the virtual recovery appliance will mount disks, format disks, and so on. When failover occurs, a new virtual machine is powered on using the replicated disks from the appliance. Once the new virtual machine is online, it will have the identity, data, and system state of the source. Since the appliance maintains its own identity, it can be reused for additional failovers.

You have the choice of using an OVF (Open Virtualization Format) virtual machine included with Carbonite Availability for your appliance, or creating your own appliance that meets the requirements below. In either case, keep in mind the following caveats for the appliance.

- The virtual recovery appliance must be a standalone virtual machine.
- It should not reside in any multiple virtual machine vApp.
- The OVF appliance is pre-configured for optimal performance. You do not need to modify the memory, CPU, or other configurations.
- You should not install or run anything else on the appliance.
- The firewall is disabled on the OVF appliance and should remain disabled.
- A single virtual recovery appliance can protect a maximum of 59 volume groups and raw block devices (combined) from any number of sources.

If you are creating your own appliance, it must meet the following requirements.

- **Operating system**—The virtual machine must be running a 64-bit version of one of the following operating systems.
 - Ubuntu 16.04.2
 - Red Hat Enterprise Linux or CentOS version 7.2, 7.3, or 7.4
 - SUSE Linux Enterprise version 12.0, 12.1, or 12.2



A SLES appliance can only protect source servers running a Carbonite Availability supported SLES version. You cannot protect other Linux operating systems to a SLES appliance.

You cannot protect Btrfs to a Red Hat or CentOS appliance.

- **Memory**—The virtual machine must have at least 4 GB of virtualized physical RAM.
- **CPUs**—The virtual machine must have at least two CPUs (two virtual sockets, not two virtual cores).
- **Disk space**—The virtual machine must have at least 16 GB of disk space available.
- **Networking**—The virtual machine must have a valid, working network configuration, including DNS.
- **Function**—The virtual machine must be dedicated to Carbonite Availability processing only. Do not use the virtual machine for any other activity (web server, database server, and so on).

- **Volume group name**—If your virtual machine is running Red Hat or CentOS and is using an LVM setup, you must make sure the volume group on the virtual machine is using a unique name. If the same volume group name is used as any volume group name from a protected source, failover will fail because of a name conflict. Refer to your Red Hat documentation for details on renaming a volume group.
- **Packages**—You will need specific packages installed on your appliance depending on the operating system of your source servers.
 - **Ext**—If the source server you will be protecting has the ext file system, you must have the e2fsprogs package on your appliance.
 - **Xfs**—If the source server you will be protecting has the xfs file system, you must have the xfsprogs package on your appliance.
 - **LVM**—If the source server you will be protecting has an LVM setup, you must have the lvm2 package on your appliance.
 - **Btrfs**—If the source server you will be protecting has the Btrfs file system and you are using an Ubuntu appliance, the appliance must have the btrfs-tools package. If the source server you are protecting is SLES 12.x with Btrfs and you are using a SLES appliance, the btrfsprogs package should already be on the SLES appliance by default. You cannot protect Btrfs to a Red Hat or CentOS appliance.
- **Permissions**—If you want to limit the permissions required for the account that you will be using for your full server to ESX job, your account must have at a minimum the permissions listed below. These permissions can be set at the vCenter, Datacenter, or host level.
 - **Datastore**—Allocate Space, Browse Datastore, Low level file operations, and Remove File
 - **Host, Local Operations**—Create Virtual Machine, Delete Virtual Machine, and Reconfigure virtual machine
 - **Network**—Assign Network
 - **Resource**—Assign virtual machine to resource pool
 - **Scheduled Task**—Create Tasks, Modify Task, Remove Task, and Run Task
 - **Tasks**—Create task and Update task
 - **Virtual Machine, Configuration**—Add existing disk, Add new disk, Add or remove device, Change resource, Modify device settings, and Remove disk
 - **Virtual Machine, Interaction**—Device connection, Power off, and Power on
 - **Virtual Machine, Inventory**—Create new, Register, Remove, and Unregister

Make sure if you also define permissions at the VMs and Templates level in vCenter that you have not denied any of the required permissions listed above.

- **System memory**—The minimum system memory on each server is 1 GB.
- **Disk space for program files**—This is the amount of disk space needed for the Carbonite Availability program files. This is approximately 400 MB on a Linux source server. The appliance needs approximately 620 MB.



Make sure you have additional disk space for Carbonite Availability queuing, logging, and so on.

- **Server name**—Carbonite Availability includes Unicode file system support, but your server name

must still be in ASCII format. Additionally, all Carbonite Availability servers and appliances must have a unique server name.

- **Target drivers**—Install on the source any drivers that are required on the target after failover. For example, you need to install on the source any NIC drivers that will be required on the target after failover.
- **Protocols and networking**—Your servers must meet the following protocol and networking requirements.
 - Your servers must have TCP/IP with static IP addressing.
 - IPv4 is the only supported version.
 - If you are using Carbonite Availability over a WAN and do not have DNS name resolution, you will need to add the host names to the local hosts file on each server running Carbonite Availability.
- **NAT support**—Carbonite Availability supports IP and port forwarding in NAT environments with the following caveats.
 - Only IPv4 is supported.
 - Only standalone servers are supported.
 - Make sure you have added your server to the Carbonite Replication Console using the correct public or private IP address. The name or IP address you use to add a server to the console is dependent on where you are running the console. Specify the private IP address of any servers on the same side of the router as the console. Specify the public IP address of any servers on the other side of the router as the console.
 - DNS failover and updates will depend on your configuration
 - Only the source or target can be behind a router, not both.
 - The DNS server must be routable from the target
- **Name resolution**—Your servers must have name resolution or DNS. The Carbonite Replication Console must be able to resolve the virtual recovery appliance, and the virtual recovery appliance must be able to resolve all source servers. For details on name resolution options, see your Linux documentation or online Linux resources.
- **Ports**—Port 1501 is used for localhost communication between the engine and management service and should be opened inbound and outbound for both TCP and UDP in iptables. Ports 1500, 1505, 1506, 6325, and 6326 are used for component communication and must be opened inbound and outbound for both TCP and UDP on any firewall that might be in use.
- **Security**—Carbonite Availability security is granted through membership in user groups. The groups can be local or LDAP (Lightweight Directory Access Protocol). A user must provide a valid local account that is a member of the Carbonite Availability security groups.
- **SELinux policy**—SELinux should be disabled on the source and target.
- **UEFI, trusted boot, secure boot**—The source boot mode cannot be UEFI (Unified Extensible Firmware Interface), trusted boot (tboot), secure boot, or other volume blocking mechanisms.
- **Docker**—Your source cannot be a Docker host.
- **Mount option**—The mount option noexec is not supported on the /tmp filesystem.
- **Snapshots**—You can take and failover to snapshots using a full server to ESX job. Because

Carbonite Availability uses VMware for snapshot capabilities, you must be aware of the requirements and limitations imposed by VMware. See VMware Knowledge Base article 1025279 at kb.vmware.com/kb/1025279 for details on best practices for VMware snapshots. Additionally, you cannot reuse a virtual disk if it has snapshots associated with it. You must delete all snapshots before you can reuse a virtual disk.

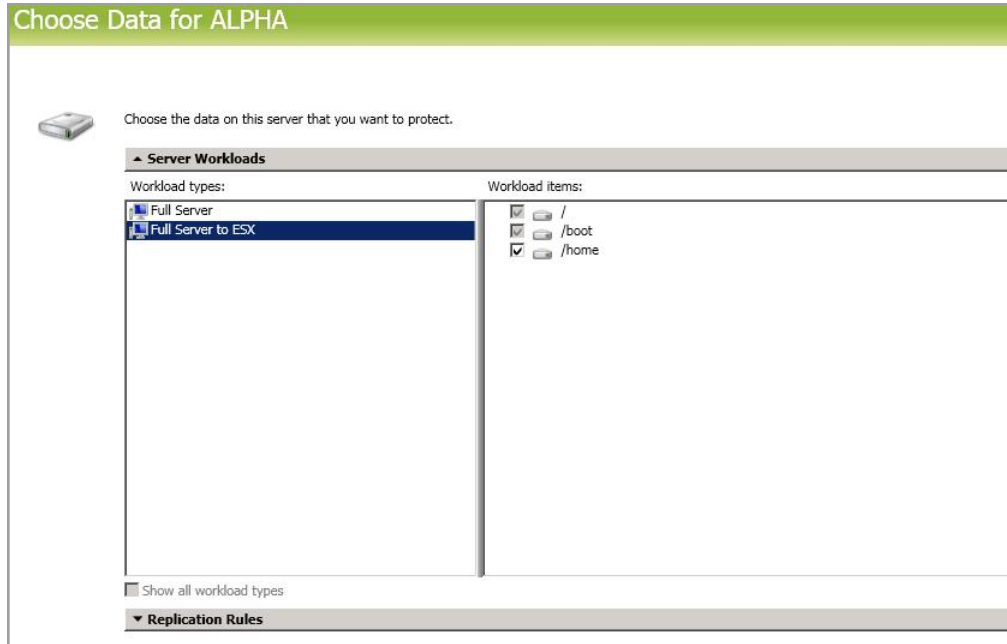
- **Supported configurations**—The following table identifies the supported configurations for a full server to ESX job.

Server to Host Configuration	Description	Supported	Not Supported
One to one active/standby	You can protect a single source to a single target host.	X	
One to one active/active	You cannot protect a single source to a single target host where the server and host act as both a source and target actively replicating data to each other.		X
Many to one	You can protect many source servers to one target host. Replication occurs from each source to the one target host. This will consolidate your source servers to a single host.	X	
One to many	You cannot protect a single source to multiple target hosts.		X
Chained	You cannot protect a single source to a single target host, where the target host then acts a source in order to send the original source to another target.		X
Single server	You cannot protect a single source to itself.		X
Standalone to standalone	Your source and target host can be in a standalone to standalone configuration.	X	
Standalone to cluster	Your source and target host cannot be in a standalone to cluster configuration.		X
Cluster to standalone	Your source and target host cannot be in a cluster to standalone configuration.		X
Cluster to cluster	Your source and target host cannot be in a cluster to cluster configuration.		X

Creating a full server to ESX job

Use these instructions to create a full server to ESX job.

1. From the **Servers** page, right-click the server you want to protect and select **Protect**. You can also highlight a server, click **Create a New Job** in the toolbar, then select **Protect**.
2. Choose the type of workload that you want to protect. Under **Server Workloads**, in the **Workload types** pane, select **Full Server to ESX**. In the **Workload items** pane, select the volumes on the source that you want to protect.



Unsupported file systems will be displayed but will not be accessible.

3. By default, Carbonite Availability selects the system and boot volumes for protection. You will be unable to deselect these volumes. Select any other volumes on the source that you want to protect.



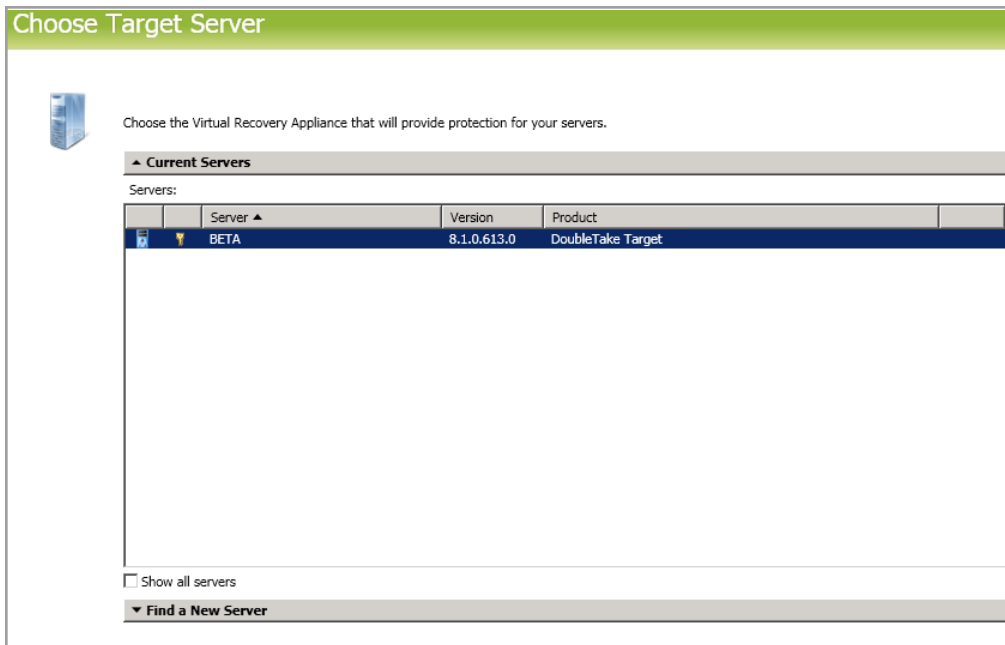
The swap partition is excluded by default and you cannot select it, however, a swap partition will be created on the replica.

If desired, click the **Replication Rules** heading and expand the volumes under **Folders**. You will see that Carbonite Availability automatically excludes particular files that cannot be used during the protection. If desired, you can exclude other files that you do not want to protect, but be careful when excluding data. Excluded volumes, folders, and/or files may compromise the integrity of your installed applications.



If you return to this page using the **Back** button in the job creation workflow, your **Workload Types** selection will be rebuilt, potentially overwriting any manual replication rules that you specified. If you do return to this page, confirm your **Workload Types** and **Replication Rules** are set to your desired settings before proceeding forward again.

4. Click **Next** to continue.
5. Choose your target server. This is the virtual recovery appliance on your ESX server.



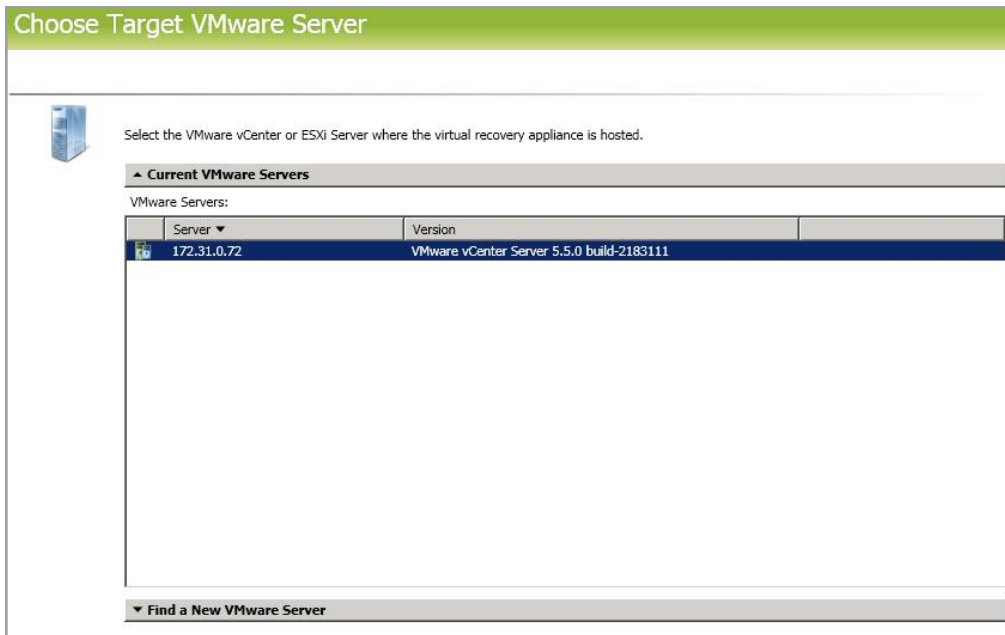
- **Current Servers**—This list contains the servers currently available in your console session. Servers that are not licensed for the workflow you have selected and those not applicable to the workload type you have selected will be filtered out of the list. Select your target server from the list. If the server you are looking for is not displayed, enable **Show all servers**. The servers in red are not available for the source server or workload type you have selected. Hover your mouse over an unavailable server to see a reason why this server is unavailable.
- **Find a New Server**—If the server you need is not in the **Current Servers** list, click the **Find a New Server** heading. From here, you can specify a server along with credentials for logging in to the server. If necessary, you can click **Browse** to select a server from a network drill-down list.



If you enter the target server's fully-qualified domain name, the Carbonite Replication Console will resolve the entry to the server short name. If that short name resides in two different domains, this could result in name resolution issues. In this case, enter the IP address of the server.

When specifying credentials for a new server, specify a user that is a member of the local dtadmin security group.

6. Click **Next** to continue.
7. Choose the server where your target virtual recovery appliance is located. This is also the server where your replica virtual machine will be located.



- **Current VMware Servers**—This list contains the vCenter and ESX servers currently available in your console session. Select your server from the list.
- **Find a New VMware Server**—If the server you need is not in the **Current VMware Servers** list, click the **Find a New VMware Server** heading.
 - **vCenter/ESXi Server**—Select your server from the list. If your server is not in the list, manually type it in.
 - **User name**—Specify the root user or another user that has the administrator role on the specified server.
 - **Password**—Specify the password associated with the **User name** you entered.
 - **Domain**—If you are working in a domain environment, specify the **Domain**.

If your server name does not match the security certificate or the security certificate has expired, you will be prompted if you want to install the untrusted security certificate.

8. Click **Next** to continue.



You may be prompted for a route from the target to the source. This route, and a port if you are using a non-default port, is used so the target can communicate with the source to build job options. This dialog box will be displayed, only if needed.

9. You have many options available for your full server to ESX job. Configure those options that are applicable to your environment.

Go to each page identified below to see the options available for that section of the **Set Options** page. After you have configured your options, continue with the next step on page 253.

- *General* on page 233
- *Replica Virtual Machine Location* on page 234
- *Replica Virtual Machine Configuration* on page 235
- *Replica Virtual Machine Volumes* on page 236
- *Replica Virtual Machine Network Settings* on page 243
- *Test Failover* on page 244
- *Failover Monitor* on page 246
- *Failover Options* on page 248
- *Mirror, Verify & Orphaned Files* on page 249
- *Network Route* on page 250
- *Snapshots* on page 251
- *Compression* on page 252
- *Bandwidth* on page 253

General



The screenshot shows a window titled "General" with a small upward-pointing arrow icon on the left. Below the title bar, there is a label "Job name:" followed by a text input field containing the text "alpha to beta".

For the **Job name**, specify a unique name for your job.

Replica Virtual Machine Location

Replica Virtual Machine Location

Select the datastore on the target ESX server that will hold the replica virtual machine:

Volume ▲	Total Size	Provisioned Space	Free Space	Owner	
EMC5	399.75 GB	141.46 GB	57.33 GB	esx51	
EMC6	399.75 GB	207.84 GB	25.88 GB	esx51	
EMC7	399.75 GB	349.34 GB	33.55 GB	esx51	

Select one of the volumes from the list to indicate the datastore on the target where you want to store the configuration files for the new virtual server when it is created. The target datastore must have enough **Free Space**. You can select the location of the .vmdk files under **Replica Virtual Machine Volumes**.

Replica Virtual Machine Configuration

	Source	Replica
Sockets	1	1
Cores per socket	1	1
Memory (MB)	4096	4096

	Source Network Adapter	Replica Virtual Switch
	eth1	InternalNetwork

- **Display name**—Specify the name of the replica virtual machine. This will be the display name of the virtual machine on the host system.
- **Hardware configuration**—Specify how you want the replica virtual machine to be created.
 - **Sockets**—Specify how many sockets to create on the new virtual machine. The number of sockets on the source is displayed to guide you in making an appropriate selection. If you select fewer sockets than the source, your clients may be impacted by slower responses.
 - **Cores per socket**—Specify how many cores to create per socket. The number of cores per socket on the source is displayed to guide you in making an appropriate selection.
 - **Memory**—Specify the amount of memory, in MB, to create on the new virtual machine. The memory on the source is displayed to guide you in making an appropriate selection. If you select less memory than the source, your clients may be impacted by slower responses.
- **Virtual switches**—Identify how you want to handle the network mapping after failover. The **Source Network Adapter** column lists the NICs from the source. Map each one to a **Replica Virtual Switch**, which is a virtual network on the target. You can also choose to discard the source's NIC and IP addresses.

Replica Virtual Machine Volumes

- **Create disks matching source**—Select this option if you want the disk configuration on the target replica to match the disk configuration on the source.



If your source has LVM, the logical volume groups on the source cannot contain physical volumes which are created based on unpartitioned disks, such as `/dev/sdb`. Instead, partitions should be created on the disks first, such as `/dev/sdb1`, and physical volumes should be created based on the partitions, before applying them to the logical volume groups. If your source physical volumes are based on unpartitioned disks, you must select the per volume configuration.

Replica Virtual Machine Volumes

Disk Configuration Strategy

Create disks matching source
 Create disks per volume

Disks:

`/dev/sda`

Disk Properties:

Virtual disk: Create new disk

Disk size: 127 GB

Datastore: EMC LUN 02

Replica disk format: Flat disk

Desired disk size: 127 GB

- **Virtual Disk**—Specify if you want Carbonite Availability to create a new disk for your replica virtual machine or if you want to use an existing disk. If you have more than one disk, you cannot mix and match new and existing. They must all be new disks or all existing disks.

Reusing a virtual disk can be useful for pre-staging data on a LAN and then relocating the virtual disk to a remote site after the initial mirror is complete. You save time by skipping the virtual disk creation steps and performing a difference mirror instead of a full mirror. With pre-staging, less data will need to be sent across the wire initially. In order to use an existing virtual disk, it must be a valid virtual disk, it cannot be attached to any other virtual machine, and it cannot have any associated snapshots.

Each pre-existing disk must be located on the target datastore specified. If you have copied the `.vmdk` file to this location manually, be sure you have also copied the associated `-flat.vmdk` file too. If you have used vCenter to copy the virtual machine, the associated file will automatically be copied. There are no restrictions on the file name of the `.vmdk`, but the associated `-flat.vmdk` file must have the same base name and the reference to that flat file in the `.vmdk` must be correct. Carbonite Availability will move, not copy, the virtual disk files to the appropriate folders created by the replica, so make sure the selected target datastore is where you want the replica virtual disk to be located.

In a WAN environment, you may want to take advantage of using an existing disk by using a process similar to the following.

- a. Create a job in a LAN environment, letting Carbonite Availability create the virtual disk for you.
- b. Complete the mirror process locally.
- c. Delete the job and when prompted, do not delete the replica.
- d. Move the virtual disk files to the desired target datastore. Do not forget to move the associated `-flat.vmdk` file if you move the files manually.
- e. Create a new protection job for the same source and reuse your existing disk.



If you have reused some existing disks and created some new disks, the numbering of the hard disks will not be identical on the source and the replica virtual machine. New disks will be created first and then existing disks will be attached. VMware assigns the hard disk numbers in order of creation and then those that are attached. The Virtual Device Node SCSI IDs will still be correct and there will be no impact within the guest of the replica virtual machine.

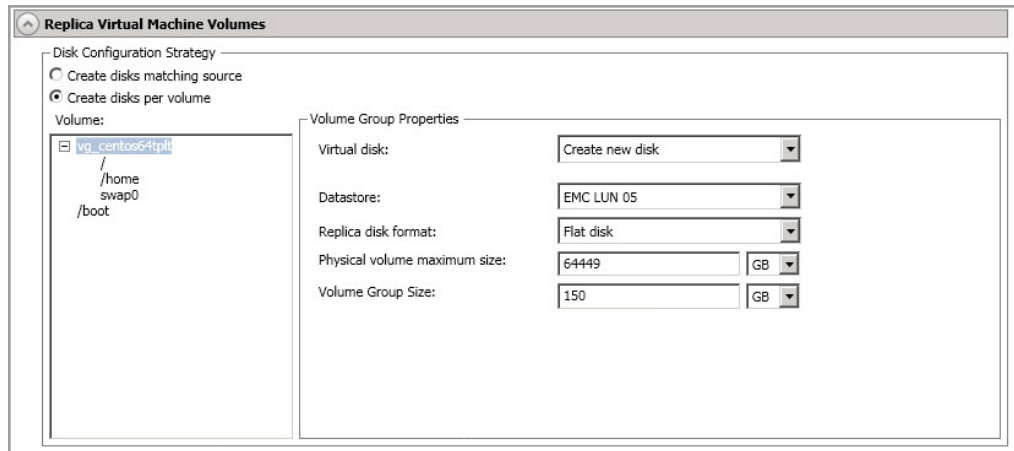
If your source has multiple partitions inside a single `.vmdk`, you can only use an existing virtual disk that Carbonite Availability created. You can only use an existing virtual disk created outside of Carbonite Availability if there is one partition in each pre-existing disk.

If you are using Logical Volume Manager, then you can only use existing disks when creating a new full server to ESX appliance job if the existing disks were created using Carbonite Availability version 7.1 or later. Versions prior to 7.1 have important LVM information deleted when the job is deleted, thus you cannot reuse the disk for a future job. If you are not using LVM, this is not an issue.

You cannot reuse a virtual disk if it has snapshots associated with it. You must delete all snapshots before you can reuse a virtual disk.

-
- **Datastore**—Specify the datastore where you want to store the `.vmdk` files for the disk. You can specify the location of the virtual machine configuration files in the **Replica Virtual Machine Location** section.
 - **Replica disk format**—If you are creating a new disk, specify the format of the disk that will be created.
 - **Flat Disk**—This disk format allocates the full amount of the disk space immediately, but does not initialize the disk space to zero until it is needed.
 - **Thick**—This disk format allocates the full amount of the disk space immediately, initializing all of the allocated disk space to zero.
 - **Thin**—This disk format does not allocate the disk space until it is needed.
 - **Desired disk size**—If you are creating a new disk, specify the maximum size, in MB or GB, of the disks.
 - **Pre-existing disk path**—If you are using an existing virtual disk, specify the location of the existing virtual disk that you want to reuse.

- **Create disks per volume**—Select this option if you want the disk configuration on the target replica to be per source volume.
 - **Volume Group Properties**—If your source has volume groups, you will see them listed in the **Volume** list. Highlight a volume group and set the available **Volume Group Properties** that are displayed to the right of the **Volume** list. The fields displayed in the **Volume Group Properties** will depend on your selection for **Virtual disk**.



- **Virtual Disk**—Specify if you want Carbonite Availability to create a new disk for your replica virtual machine or if you want to use an existing disk.

Reusing a virtual disk can be useful for pre-staging data on a LAN and then relocating the virtual disk to a remote site after the initial mirror is complete. You save time by skipping the virtual disk creation steps and performing a difference mirror instead of a full mirror. With pre-staging, less data will need to be sent across the wire initially. In order to use an existing virtual disk, it must be a valid virtual disk, it cannot be attached to any other virtual machine, and it cannot have any associated snapshots.

Each pre-existing disk must be located on the target datastore specified. If you have copied the .vmdk file to this location manually, be sure you have also copied the associated -flat.vmdk file too. If you have used vCenter to copy the virtual machine, the associated file will automatically be copied. There are no restrictions on the file name of the .vmdk, but the associated -flat.vmdk file must have the same base name and the reference to that flat file in the .vmdk must be correct. Carbonite Availability will move, not copy, the virtual disk files to the appropriate folders created by the replica, so make sure the selected target datastore is where you want the replica virtual disk to be located.

In a WAN environment, you may want to take advantage of using an existing disk by using a process similar to the following.

- Create a job in a LAN environment, letting Carbonite Availability create the virtual disk for you.
- Complete the mirror process locally.
- Delete the job and when prompted, do not delete the replica.

- d. Move the virtual disk files to the desired target datastore. Do not forget to move the associated `-flat.vmdk` file if you move the files manually.
- e. Create a new protection job for the same source and reuse your existing disk.



If you have reused some existing disks and created some new disks, the numbering of the hard disks will not be identical on the source and the replica virtual machine. New disks will be created first and then existing disks will be attached. VMware assigns the hard disk numbers in order of creation and then those that are attached. The Virtual Device Node SCSI IDs will still be correct and there will be no impact within the guest of the replica virtual machine.

If your source has multiple partitions inside a single `.vmdk`, you can only use an existing virtual disk that Carbonite Availability created. You can only use an existing virtual disk created outside of Carbonite Availability if there is one partition in each pre-existing disk.

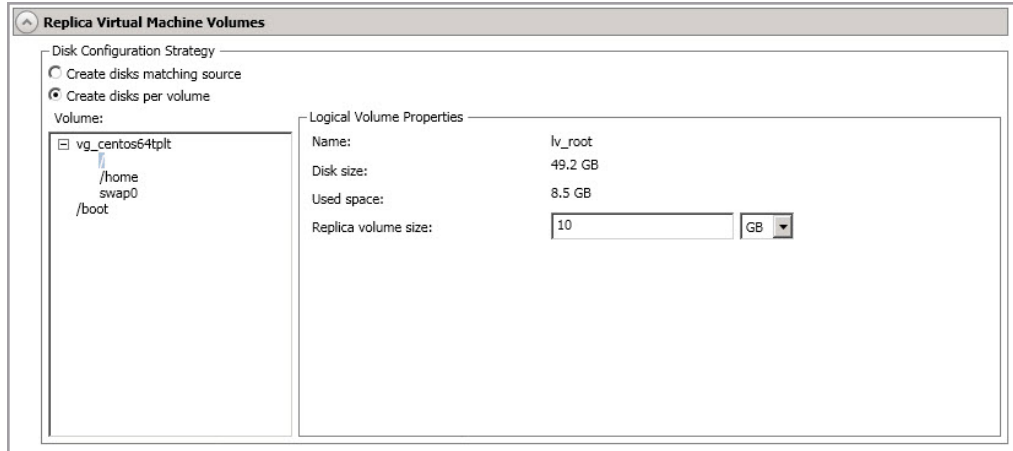
If you are using Logical Volume Manager, then you can only use existing disks when creating a new full server to ESX appliance job if the existing disks were created using Carbonite Availability version 7.1 or later. Versions prior to 7.1 have important LVM information deleted when the job is deleted, thus you cannot reuse the disk for a future job. If you are not using LVM, this is not an issue.

You cannot reuse a virtual disk if it has snapshots associated with it. You must delete all snapshots before you can reuse a virtual disk.

- **Datastore**—Specify the datastore where you want to store the `.vmdk` files for the volume group. You can specify the location of the virtual machine configuration files in the **Replica Virtual Machine Location** section.
- **Replica disk format**—If you are creating a new disk, specify the format of the disk that will be created.
 - **Flat Disk**—This disk format allocates the full amount of the disk space immediately, but does not initialize the disk space to zero until it is needed.
 - **Thick**—This disk format allocates the full amount of the disk space immediately, initializing all of the allocated disk space to zero.
 - **Thin**—This disk format does not allocate the disk space until it is needed.
- **Physical volume maximum size**—If you are creating a new disk, specify the maximum size, in MB or GB, of the virtual disks used to create the volume group. The default value is equal to the maximum size that can be attached to the datastore you selected. That will depend on your ESX version, your file system version, and the block size of your datastore.
- **Volume Group size**—If you are creating a new disk, specify the maximum size, in MB or GB, of the volume group. The default value will match the

source. This value cannot be less than the logical volumes total size that you are trying to create on the volume group.

- **Pre-existing virtual disks path**—If you are using an existing virtual disk, specify the location of the existing virtual disks that you want to reuse.
- **Logical Volume Properties**—If your source has logical volumes, you will see them listed in the **Volume** list. Highlight a logical volume and set the available **Logical Volume Properties** that are displayed to the right of the **Volume** list.



If you are using an existing virtual disk, you will not be able to modify the logical volume properties.

The size and space displayed may not match the output of the Linux `df` command. This is because `df` shows the size of the mounted file system not the underlying partition which may be larger. Additionally, Carbonite Availability uses powers of 1024 when computing GB, MB, and so on. The `df` command typically uses powers of 1000 and rounds up to the nearest whole value.

- **Name**—This field displays the logical volume name.
- **Disk size**—This field displays the size of the logical volume on the source.
- **Used space**—This field displays the amount of disk space in use on the source logical volume.
- **Replica volume size**—Specify the size, in MB or GB, of the replica logical volume on the target. The value must be at least the size of the specified **Used space** on that volume.



In some cases, the replica virtual machine may use more virtual disk space than the size of the source volume due to differences in how the virtual disk's block size is formatted and how hard links are handled. To avoid this issue, specify the size of your replica to be at least 5 GB



larger.

- **Partition Properties**—If your source has partitions, you will see them listed in the **Volume** list. Highlight a partition and set the available **Partition Properties** that are displayed to the right of the **Volume** list. The fields displayed in the **Partition Properties** will depend on your selection for **Virtual disk**.



The size and space displayed may not match the output of the Linux `df` command. This is because `df` shows the size of the mounted file system not the underlying partition which may be larger. Additionally, Carbonite Availability uses powers of 1024 when computing GB, MB, and so on. The `df` command typically uses powers of 1000 and rounds up to the nearest whole value.

- **Virtual Disk**—Specify if you want Carbonite Availability to create a new disk for your replica virtual machine or if you want to use an existing disk. Review the details above under **Volume Group Properties Virtual Disk** for information on using an existing disk.
- **Disk size**—This field displays the size of the partition on the source.
- **Used space**—This field displays the amount of disk space in use on the source partition.
- **Datastore**—Specify the datastore where you want to store the `.vmdk` files for the partition. You can specify the location of the virtual machine configuration files in the **Replica Virtual Machine Location** section.
- **Replica disk format**—Specify the format of the disk that will be created.
 - **Flat Disk**—This disk format allocates the full amount of the disk space immediately, but does not initialize the disk space to zero until it is needed.
 - **Thick**—This disk format allocates the full amount of the disk space immediately, initializing all of the allocated disk space to zero.

- **Thin**—This disk format does not allocate the disk space until it is needed.
- **Replica volume size**—Specify the size, in MB or GB, of the replica partition on the target. The value must be at least the size of the specified **Used space** on that partition.
- **Pre-existing disks path**—If you are using an existing virtual disk, specify the location of the existing virtual disks that you want to reuse.

Replica Virtual Machine Network Settings

Use advanced settings for replica virtual machine network configuration.

Network adapters:

eth0 (112.42.74.29)

Source IP addresses:

IP Address	Subnet Mask
112.42.74.29	255.255.0.0

Replica IP addresses:

IP Address	Subnet Mask
112.52.74.29	255.255.0.0

Source Default Gateways:

112.42.48.9

Replica Default Gateways:

112.52.48.9

Source DNS Server addresses:

112.42.48.20

Replica DNS Server addresses:

112.52.48.20

- **Use advanced settings for replica virtual machine network configuration**—Select this option to enable the replica virtual machine network setting configuration. This setting is primarily used for WAN support.
- **Network adapters**—Select a network adapter from the source and specify the **Replica IP addresses**, **Replica Default Gateways**, and **Replica DNS Server addresses** to be used after failover. If you add multiple gateways or DNS servers, you can sort them by using the arrow up and arrow down buttons. Repeat this step for each network adapter on the source.



Updates made during failover will be based on the network adapter name when protection is established. If you change that name, you will need to delete the job and re-create it so the new name will be used during failover.

If you update one of the advanced settings (IP address, gateway, or DNS server), then you must update all of them. Otherwise, the remaining items will be left blank. If you do not specify any of the advanced settings, the replica virtual machine will be assigned the same network configuration as the source.

By default, the source IP address will be included in the target IP address list as the default address. If you do not want the source IP address to be the default address on the target after failover, remove that address from the **Replica IP addresses** list.

Linux operating systems only support one gateway, so the first gateway listed will be used.

Test Failover

These options allow you to perform a test failover. Keep in mind the following for using test failover.



Test failover is not supported for Btrfs file systems.

In order to use test failover, you must make sure your target has sufficient free disk space available. The free space on each volume group on the target must be larger than 50% of the total size of all logical volumes in that volume group on the source. Meeting that amount of free space may depend on the **Disk Configuration Strategy** you selected under **Replica Virtual Disk Volumes**.

- **Create disks matching source**—With the match source option, you must have sufficient free space on the source before the job is created because the target disks will be matching the source. You may need to increase free disk space on the source (perhaps add a partition which has been created on the raw disk on the source and then extend the volume group), in order to allow for sufficient free space to be matched on the target.
 - **Create disks per volume**—With the per volume option, select a volume group size on the target that has enough free space to accommodate the 50% free space requirement.
-
- The source, target, and protection job will remain online and uninterrupted during the test.
 - The test will be performed using the test failover settings configured during job creation.
 - The test will use the current data on the target.
 - Scheduled snapshots will be deferred during the test and taken automatically after the test is undone.
 - The test failover will take a snapshot of the current data on the target and create a new set of virtual disks. The data from the snapshot will be mirrored to the new set of disks using the same mirroring options as the protection job.
 - Once the mirror is complete, the replica virtual machine or alternate replica virtual machine, depending on your selected configuration, is automatically brought online using the new set of disks.
 - The replica virtual machine or alternate replica virtual machine, depending on your selected configuration, will use the network settings specified in the test failover settings of the protection job.
 - When you are finished with your test, undo it.
 - When you undo a test failover, the new set of disks will be maintained or deleted as specified in the test failover settings of the protection job.
 - At any time during a test failover, you can undo the test, perform a live failover, or failover to a snapshot. (Performing a live failover or failing over to a snapshot will automatically undo any test in progress.)

Test Failover

Virtual Machine

Use default replica virtual machine

Use alternate test replica virtual machine

Display name:

Network

Do not connect replica network adapters on test failover

Connect and map replica network adapters on test failover

Map source virtual switches to target virtual switches for test failover:

Source Network Adapter	Replica Virtual Switch
eth1	Internal

Delete test failover virtual disks

- **Use default replica virtual machine**—Select this option to use the same replica virtual machine that will be used for live failover for the test failover. Do not use this option if you are using the snapshot functionality on the target. If there are snapshots on the target and you are using the default replica virtual machine for test failover, the test will fail and an error message will be logged.
- **Use alternate test replica virtual machine**—Select this option to create an alternate replica virtual machine to use for the test failover. You must use this option if you want to also use the snapshot functionality on the target.
- **Display name**—Specify the name of the alternate replica virtual machine to use for the test. This will be the display name of the virtual machine on the host system.
- **Do not connect replica network adapters on test failover**—Select this option if you do not want the replica virtual machine used for the test to be connected to the network.
- **Connect and map replica network adapters on test failover**—Select this option if you want the replica virtual machine used for the test to be connected to the network. You will need to map each **Source Network Adapter** to a **Target Virtual Switch** for the test. You can also choose to discard the source's NIC and IP addresses.
- **Delete test failover virtual disks**—Select this option if you want to delete the new virtual disks created during the test failover process. If you disable this option, the new disks will not be deleted when you perform undo failover.



Be careful if you choose to connect the network adapters for a test failover. Depending on your network adapter mappings, users may be able to access the target. Also, since the source is still online, there is a chance users may split between accessing the source or target.

Failover Monitor

Failover Monitor

Total time to failure: 00:05:00

Consecutive failures: 20

Monitor on this interval: 00:00:10

Network monitoring

Monitor these addresses:

Source IP Address
<input checked="" type="checkbox"/> 172.31.206.201

Monitoring method: Network service

Failover trigger: All monitored IP addresses fail

- **Total time to failure**—Specify, in hours:minutes:seconds, how long the target will keep trying to contact the source before the source is considered failed. This time is precise. If the total time has expired without a successful response from the source, this will be considered a failure.

Consider a shorter amount of time for servers, such as a web server or order processing database, which must remain available and responsive at all times. Shorter times should be used where redundant interfaces and high-speed, reliable network links are available to prevent the false detection of failure. If the hardware does not support reliable communications, shorter times can lead to premature failover. Consider a longer amount of time for machines on slower networks or on a server that is not transaction critical. For example, failover would not be necessary in the case of a server restart.

- **Consecutive failures**—Specify how many attempts the target will make to contact the source before the source is considered failed. For example, if you have this option set to 20, and your source fails to respond to the target 20 times in a row, this will be considered a failure.
- **Monitor on this interval**—Specify, in hours:minutes:seconds, how long to wait between attempts to contact the source to confirm it is online. This means that after a response (success or failure) is received from the source, Carbonite Availability will wait the specified interval time before contacting the source again. If you set the interval to 00:00:00, then a new check will be initiated immediately after the response is received.

If you choose **Total time to failure**, do not specify a longer interval than failure time or your server will be considered failed during the interval period.

- **Network monitoring**—With this option, the target will monitor the source using a network ping.
 - **Monitor these addresses**—Select each **Source IP Address** that you want the target to monitor. If you are using a NAT environment, do not select a private IP address on the source because the target cannot reach the source's private address,

thus causing an immediate failure. Also for NAT environments, you will see another column for the **Replication Service port**. This gives you the ability to specify the port number to be used with the address, allowing the target to monitor the source through a router.

- **Monitoring method**—This option determines the type of network ping used for failover monitoring.
 - **Network service**—Source availability will be tested by an ICMP ping to confirm the route is active.
 - **Replication service**—Source availability will be tested by a UDP ping to confirm the Double-Take service is active.
 - **Network and replication services**—Source availability will be tested by both an ICMP ping to confirm the route is active and a UDP ping to confirm the Double-Take service is active. If either monitoring method fails, failover will be triggered.
- **Failover trigger**—If you are monitoring multiple IP addresses, specify when you want a failover condition to be triggered.
 - **One monitored IP address fails**—A failover condition will be triggered when any one of the monitored IP addresses fails. If each IP address is on a different subnet, you may want to trigger failover after one fails.
 - **All monitored IP addresses fail**—A failover condition will be triggered when all monitored IP addresses fail. If there are multiple, redundant paths to a server, losing one probably means an isolated network problem and you should wait for all IP addresses to fail.

Failover Options

The screenshot shows a window titled "Failover Options" with a close button in the top-left corner. Inside the window, there is a checked checkbox labeled "Wait for user to initiate failover". Below this, there is a section titled "Target scripts" which contains two rows. The first row is for "Pre-failover script:" and the second is for "Post-failover script:". Each row has a text input field followed by a three-dot menu icon and an "Arguments:" label with another text input field. A checkbox labeled "Delay failover until script completes" is located between the two script rows.

- **Wait for user to initiate failover**—The failover process can wait for you to initiate it, allowing you to control when failover occurs. When a failure occurs, the job will wait in **Failover Condition Met** for you to manually initiate the failover process. Disable this option if you want failover to occur immediately when a failure occurs.
- **Target Scripts**—You can customize failover by running scripts on the target appliance or the replica. Scripts may contain any valid Linux command, executable, or shell script file. The scripts are processed using the same account running the Double-Take Management service. Examples of functions specified in scripts include stopping and starting services, stopping and starting applications or processes, notifying the administrator before and after failover occurs, and so on. There are two types of failover scripts.
 - **Pre-failover script**—This script runs on the target appliance at the beginning of the failover process. Specify the full path and name of the script file.
 - **Delay until script completes**—Enable this option if you want to delay the failover process until the associated script has completed. If you select this option, make sure your script handles errors, otherwise the failover process may never complete if the process is waiting on a script that cannot complete.
 - **Post-failover script**—This script runs on the replica at the end of the failover process. Specify the full path and name of the script file.
 - **Arguments**—Specify a comma-separated list of valid arguments required to execute the script.

Mirror, Verify & Orphaned Files

Mirror, Verify & Orphaned Files

Mirror Options

Choose a comparison method and whether to mirror the entire file or only the bytes that differ in each file.

Compare file attributes. Send the attributes and bytes that differ.

General Options

Delete orphaned files

- **Mirror Options**—Choose a comparison method and whether to mirror the entire file or only the bytes that differ in each file.
 - **Do not compare files. Send the entire file.**—Carbonite Availability will not perform any comparisons between the files on the source and target. All files will be mirrored to the target, sending the entire file.
 - **Compare file attributes. Send the attributes and bytes that differ.**—Carbonite Availability will compare file attributes and will mirror only the attributes and bytes that are different.
 - **Compare file attributes and data. Send the attributes and bytes that differ.**—Carbonite Availability will compare file attributes and the file data and will mirror only the attributes and bytes that are different.
- **General Options**—Choose your general mirroring options.
 - **Delete orphaned files**—An orphaned file is a file that exists in the replica data on the target, but does not exist in the protected data on the source. This option specifies if orphaned files should be deleted on the target.



Orphaned file configuration is a per target configuration. All jobs to the same target will have the same orphaned file configuration.

If delete orphaned files is enabled, carefully review any replication rules that use wildcard definitions. If you have specified wildcards to be excluded from protection, files matching those wildcards will also be excluded from orphaned file processing and will not be deleted from the target. However, if you have specified wildcards to be included in your protection, those files that fall outside the wildcard inclusion rule will be considered orphaned files and will be deleted from the target.

Network Route



The screenshot shows a configuration window titled "Network Route". Inside the window, there is a label "Send data to the target server using this route:" followed by a dropdown menu. The dropdown menu currently displays the IP address "10.10.10.30".

By default, Carbonite Availability will select a target route for transmissions. If desired, specify an alternate route on the target that the data will be transmitted through. This allows you to select a different route for Carbonite Availability traffic. For example, you can separate regular network traffic and Carbonite Availability traffic on a machine with multiple IP addresses. You can also select or manually enter a public IP address (which is the public IP address of the server's router) if you are using a NAT environment. If you enter a public IP addresses, you will see additional fields allowing you to disable the default communication ports and specify other port numbers to use, allowing the target to communicate through a router. The **Management Service port** can be disregarded. It is used for other job types. The **Replication Service port** is used for data transmission.

Snapshots

Snapshots

Enable scheduled snapshots

Take snapshots on this interval: 1 Hours

Maximum number of snapshots: 1

Begin immediately

Begin at this time: 6/25/2015 10:58:08 AM

A snapshot is an image of the source replica data on the target taken at a single point in time. You can failover to a snapshot. However, you cannot access the snapshot to recover specific files or folders.

Turn on **Enable scheduled snapshots** if you want Carbonite Availability to take snapshots automatically at set intervals.

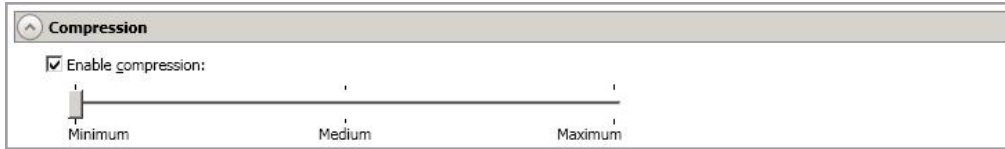
- **Take snapshots on this interval**—Specify the interval (in days, hours, or minutes) for taking snapshots. Due to VMware processing speed, you should not schedule at less than 10 minute intervals.
- **Maximum number of snapshots**—Specify the maximum number of snapshots to retain. The upper limit is 30 maximum snapshots. Once this limit is reached, a new snapshot will delete the oldest snapshot.
- **Begin immediately**—Select this option if you want to start taking snapshots immediately after the protection job is established.
- **Begin at this time**—Select this option if you want to start taking snapshots at a later date and time. Specify the date and time parameters to indicate when you want to start.



See *Managing snapshots* on page 78 for details on taking manual snapshots and deleting snapshots.

Make sure you have reviewed the snapshots best practices as noted in the *Full server to ESX requirements* on page 223.

Compression



To help reduce the amount of bandwidth needed to transmit Carbonite Availability data, compression allows you to compress data prior to transmitting it across the network. In a WAN environment this provides optimal use of your network resources. If compression is enabled, the data is compressed before it is transmitted from the source. When the target receives the compressed data, it decompresses it and then writes it to disk. You can set the level from **Minimum** to **Maximum** to suit your needs.

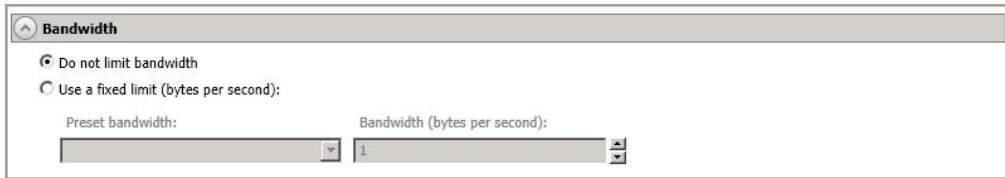
Keep in mind that the process of compressing data impacts processor usage on the source. If you notice an impact on performance while compression is enabled in your environment, either adjust to a lower level of compression, or leave compression disabled. Use the following guidelines to determine whether you should enable compression.

- If data is being queued on the source at any time, consider enabling compression.
- If the server CPU utilization is averaging over 85%, be cautious about enabling compression.
- The higher the level of compression, the higher the CPU utilization will be.
- Do not enable compression if most of the data is inherently compressed. Many image (.jpg, .gif) and media (.wmv, .mp3, .mpg) files, for example, are already compressed. Some images files, such as .bmp and .tif, are decompressed, so enabling compression would be beneficial for those types.
- Compression may improve performance even in high-bandwidth environments.
- Do not enable compression in conjunction with a WAN Accelerator. Use one or the other to compress Carbonite Availability data.



All jobs from a single source connected to the same IP address on a target will share the same compression configuration.

Bandwidth



Bandwidth limitations are available to restrict the amount of network bandwidth used for Carbonite Availability data transmissions. When a bandwidth limit is specified, Carbonite Availability never exceeds that allotted amount. The bandwidth not in use by Carbonite Availability is available for all other network traffic.



All jobs from a single source connected to the same IP address on a target will share the same bandwidth configuration.

- **Do not limit bandwidth**—Carbonite Availability will transmit data using 100% bandwidth availability.
 - **Use a fixed limit**—Carbonite Availability will transmit data using a limited, fixed bandwidth. Select a **Preset bandwidth** limit rate from the common bandwidth limit values. The **Bandwidth** field will automatically update to the bytes per second value for your selected bandwidth. This is the maximum amount of data that will be transmitted per second. If desired, modify the bandwidth using a bytes per second value. The minimum limit should be 3500 bytes per second.
10. Click **Next** to continue.
 11. Carbonite Availability validates that your source and target are compatible. The **Summary** page displays your options and validation items.

Errors are designated by a white X inside a red circle. Warnings are designated by a black exclamation point (!) inside a yellow triangle. A successful validation is designated by a white checkmark inside a green circle. You can sort the list by the icon to see errors, warnings, or successful validations together. Click on any of the validation items to see details. You must correct any errors before you can continue. Depending on the error, you may be able to click **Fix** or **Fix All** and let Carbonite Availability correct the problem for you. For those errors that Carbonite Availability cannot correct automatically, you will need to modify the source or target to correct the error, or you can select a different target. You must revalidate the selected servers, by clicking **Recheck**, until the validation check passes without errors.

If you receive a path transformation error during job validation indicating a volume does not exist on the target server, even though there is no corresponding data being protected on the source, you will need to manually modify your replication rules. Go back to the **Choose Data** page and under the **Replication Rules**, locate the volume from the error message. Remove any rules associated with that volume. Complete the rest of the workflow and the validation should pass.

After a job is created, the results of the validation checks are logged to the job log. See the *Carbonite Availability and Carbonite Move Reference Guide* for details on the various Carbonite Availability log files.

12. Once your servers have passed validation and you are ready to establish protection, click **Finish**,

and you will automatically be taken to the **Jobs** page.



Jobs in a NAT environment may take longer to start.

Once a job is created, do not change the name of underlying hardware components used in the job. For example, volume and datastore names or network adapter and virtual switch names. Any component used by name in your job must continue to use that name throughout the lifetime of job. If you must change a name, you will need to delete the job and re-create it using the new component name.

Managing and controlling full server to ESX jobs

Click **Jobs** from the main Carbonite Replication Console toolbar. The **Jobs** page allows you to view status information about your jobs. You can also control your jobs from this page.

The jobs displayed in the right pane depend on the server group folder selected in the left pane. Every job for each server in your console session is displayed when the **Jobs on All Servers** group is selected. If you have created and populated server groups (see *Managing servers* on page 55), then only the jobs associated with the server or target servers in that server group will be displayed in the right pane.

- *Overview job information displayed in the top right pane* on page 255
- *Detailed job information displayed in the bottom right pane* on page 258
- *Job controls* on page 260


Overview job information displayed in the top right pane


The top pane displays high-level overview information about your jobs. You can sort the data within a column in ascending and descending order. You can also move the columns to the left or right of each other to create your desired column order. The list below shows the columns in their default left to right order.


If you are using server groups, you can filter the jobs displayed in the top right pane by expanding the **Server Groups** heading and selecting a server group.


Column 1 (Blank)

The first blank column indicates the state of the job.

 A green circle with a white checkmark indicates the job is in a healthy state. No action is required.

 A yellow triangle with a black exclamation point indicates the job is in a pending or warning state. This icon is also displayed on any server groups that you have created that contain a job in a pending or warning state. Carbonite Availability is working or waiting on a pending process or attempting to resolve the warning state.

 A red circle with a white X indicates the job is in an error state. This icon is also displayed on any server groups that you have created that contain a job in an error state. You will need to investigate and resolve the error.

 The job is in an unknown state.

Job

The name of the job

Source Server

The name of the source. This could be the name or IP address of your source.

Target Server

The name of the target. This could be the name or IP address of your target.

Job Type

Each job type has a unique job type name. This job is a Full Server to ESX job. For a complete list of all job type names, press F1 to view the Carbonite Replication Console online help.

Activity

There are many different **Activity** messages that keep you informed of the job activity. Most of the activity messages are informational and do not require any administrator interaction. If you see error messages, check the job details. Keep in mind that **Idle** indicates console to server activity is idle, not that your servers are idle.

Mirror Status

- **Calculating**—The amount of data to be mirrored is being calculated.
- **In Progress**—Data is currently being mirrored.
- **Waiting**—Mirroring is complete, but data is still being written to the target.
- **Idle**—Data is not being mirrored.
- **Paused**—Mirroring has been paused.
- **Stopped**—Mirroring has been stopped.
- **Removing Orphans**—Orphan files on the target are being removed or deleted depending on the configuration.
- **Verifying**—Data is being verified between the source and target.
- **Unknown**—The console cannot determine the status.

Replication Status

- **Replicating**—Data is being replicated to the target.
- **Ready**—There is no data to replicate.
- **Pending**—Replication is pending.
- **Stopped**—Replication has been stopped.
- **Out of Memory**—Replication memory has been exhausted.
- **Failed**—The Double-Take service is not receiving replication operations from the Carbonite Availability driver. Check the Event Viewer for driver related issues.
- **Unknown**—The console cannot determine the status.

Transmit Mode

- **Active**—Data is being transmitted to the target.
- **Paused**—Data transmission has been paused.
- **Scheduled**—Data transmission is waiting on schedule criteria.
- **Stopped**—Data is not being transmitted to the target.
- **Error**—There is a transmission error.
- **Unknown**—The console cannot determine the status.

Operating System

The job type operating system

Detailed job information displayed in the bottom right pane

The details displayed in the bottom pane provide additional information for the job highlighted in the top pane. You can expand or collapse the bottom pane by clicking on the **Job Highlights** heading.

Name

The name of the job

Target data state

- **OK**—The data on the target is in a good state.
- **Mirroring**—The target is in the middle of a mirror process. The data will not be in a good state until the mirror is complete.
- **Mirror Required**—The data on the target is not in a good state because a remirror is required. This may be caused by an incomplete or stopped mirror or an operation may have been dropped on the target.
- **Busy**—The source is low on memory causing a delay in getting the state of the data on the target.
- **Not Loaded**—Carbonite Availability target functionality is not loaded on the target server. This may be caused by a license key error.
- **Not Ready**—The Linux drivers have not yet completed loading on the target.
- **Unknown**—The console cannot determine the status.

Mirror remaining

The total number of mirror bytes that are remaining to be sent from the source to the target.

Mirror skipped

The total number of bytes that have been skipped when performing a difference. These bytes are skipped because the data is not different on the source and target.

Replication queue

The total number of replication bytes in the source queue

Disk queue

The amount of disk space being used to queue data on the source

Recovery point latency

The length of time replication is behind on the target compared to the source. This is the time period of replication data that would be lost if a failure were to occur at the current time. This value represents replication data only and does not include mirroring data. If you are mirroring and failover, the data on the target will be at least as far behind as the replication point latency. It could potentially be further behind depending on the circumstances of the mirror. If mirroring is idle and you failover, the data will only be as far behind as the replication point latency time.

Bytes sent

The total number of mirror and replication bytes that have been transmitted to the target

Bytes sent (compressed)

The total number of compressed mirror and replication bytes that have been transmitted to the target. If compression is disabled, this statistic will be the same as **Bytes sent**.

Connected since

The date and time indicating when the current job was started.

Recent activity

Displays the most recent activity for the selected job, along with an icon indicating the success or failure of the last initiated activity. Click the link to see a list of recent activities for the selected job. You can highlight an activity in the list to display additional details about the activity.

Additional information

Depending on the current state of your job, you may see additional information displayed to keep you informed about the progress and status of your job. If there is no additional information, you will see (None) displayed.

Job controls

You can control your job through the toolbar buttons available on the **Jobs** page. If you select multiple jobs, some of the controls will apply only to the first selected job, while others will apply to all of the selected jobs. For example, **View Job Details** will only show details for the first selected job, while **Stop** will stop protection for all of the selected jobs.

If you want to control just one job, you can also right click on that job and access the controls from the pop-up menu.

View Job Details

This button leaves the **Jobs** page and opens the **View Job Details** page.

Edit Job Properties

This button leaves the **Jobs** page and opens the **EditJob Properties** page.

Delete

Stops (if running) and deletes the selected jobs.

If you no longer want to protect the source and no longer need the replica of the source on the target, select to delete the associated replica virtual machine. Selecting this option will remove the job and completely delete the replica virtual machine on the target. Do not select this option if you want to keep the replica of the source on the target. If you do not select the delete option, the source replica will be preserved on the target.

Provide Credentials

Changes the login credentials that the job (which is on the target machine) uses to authenticate to the servers in the job. This button opens the Provide Credentials dialog box where you can specify the new account information and which servers you want to update. See *Providing server credentials* on page 66. You will remain on the **Jobs** page after updating the server credentials. If your servers use the same credentials, make sure you also update the credentials on the **Servers** page so that the Carbonite Replication Console can authenticate to the servers in the console session. See *Managing servers* on page 55.

View Recent Activity

Displays the recent activity list for the selected job. Highlight an activity in the list to display additional details about the activity.

Start

Starts or resumes the selected jobs.

If you have previously stopped protection, the job will restart mirroring and replication.

If you have previously paused protection, the job will continue mirroring and replication from where it left off, as long as the Carbonite Availability queue was not exhausted during the time the job was paused. If the Carbonite Availability queue was exhausted during the time the job was paused, the job will restart mirroring and replication.

Also if you have previously paused protection, all jobs from the same source to the same IP address on the target will be resumed.

Pause

Pauses the selected jobs. Data will be queued on the source while the job is paused.

All jobs from the same source to the same IP address on the target will be paused.

Stop

Stops the selected jobs. The jobs remain available in the console, but there will be no mirroring or replication data transmitted from the source to the target. Mirroring and replication data will not be queued on the source while the job is stopped, requiring a remirror when the job is restarted. The type of remirror will depend on your job settings.

Take Snapshot

Even if you have scheduled the snapshot process, you can run it manually any time. If an automatic or scheduled snapshot is currently in progress, Carbonite Availability will wait until that one is finished before taking the manual snapshot.

Manage Snapshots

Allows you to manage your snapshots by taking and deleting snapshots for the selected job. See *Managing snapshots* on page 78 for more information.

Failover or Cutover

Starts the failover process. See *Failing over full server to ESX jobs* on page 272 for the process and details of failing over a full server to ESX job.

Failback

Starts the failback process. Failback does not apply to full server to ESX jobs.

Restore

Starts the restoration process. Restore does not apply to full server to ESX jobs.

Reverse

Reverses protection. Reverse protection does not apply to full server to ESX jobs.

Undo Failover or Cutover

Cancels a test failover by undoing it. Undo failover does not apply to full server to ESX jobs.

View Job Log

Opens the job log. On the right-click menu, this option is called **View Logs**, and you have the option of opening the job log, source server log, or target server log.

Other Job Actions

Opens a small menu of other job actions. These job actions are not available for Linux jobs.

Filter

Select a filter option from the drop-down list to only display certain jobs. You can display **Healthy jobs**, **Jobs with warnings**, or **Jobs with errors**. To clear the filter, select **All jobs**. If you have created and populated server groups, then the filter will only apply to the jobs associated with the server or target servers in that server group. See *Managing servers* on page 55.

Search

Allows you to search the source or target server name for items in the list that match the criteria you have entered.

Overflow Chevron

Displays any toolbar buttons that are hidden from view when the window size is reduced.

Viewing full server to ESX job details

From the **Jobs** page, highlight the job and click **View Job Details** in the toolbar.

Review the following table to understand the detailed information about your job displayed on the **View Job Details** page.





Job name

The name of the job

Job type

Each job type has a unique job type name. This job is a Full Server to ESX job. For a complete list of all job type names, press F1 to view the Carbonite Replication Console online help.

Health

-  The job is in a healthy state.
-  The job is in a warning state.
-  The job is in an error state.
-  The job is in an unknown state.

Activity

There are many different **Activity** messages that keep you informed of the job activity. Most of the activity messages are informational and do not require any administrator interaction. If you see error messages, check the rest of the job details.

Connection ID

The incremental counter used to number connections. The number is incremented when a connection is created. The counter is reset if there are no existing jobs and the Double-Take service is restarted.

Transmit mode

- **Active**—Data is being transmitted to the target.
- **Paused**—Data transmission has been paused.
- **Scheduled**—Data transmission is waiting on schedule criteria.
- **Stopped**—Data is not being transmitted to the target.
- **Error**—There is a transmission error.
- **Unknown**—The console cannot determine the status.

Target data state

- **OK**—The data on the target is in a good state.
- **Mirroring**—The target is in the middle of a mirror process. The data will not be in a good state until the mirror is complete.
- **Mirror Required**—The data on the target is not in a good state because a remirror is required. This may be caused by an incomplete or stopped mirror or an operation may have been dropped on the target.
- **Busy**—The source is low on memory causing a delay in getting the state of the data on the target.
- **Not Loaded**—Carbonite Availability target functionality is not loaded on the target server. This may be caused by a license key error.
- **Not Ready**—The Linux drivers have not yet completed loading on the target.
- **Unknown**—The console cannot determine the status.

Target route

The IP address on the target used for Carbonite Availability transmissions.

Compression

- **On / Level**—Data is compressed at the level specified.
- **Off**—Data is not compressed.

Encryption

- **On**—Data is being encrypted before it is sent from the source to the target.
- **Off**—Data is not being encrypted before it is sent from the source to the target.

Bandwidth limit

If bandwidth limiting has been set, this statistic identifies the limit. The keyword **Unlimited** means there is no bandwidth limit set for the job.

Connected since

The source server date and time indicating when the current job was started. This field is blank, indicating that a TCP/IP socket is not present, when the job is waiting on transmit options or if the transmission has been stopped. This field will maintain the date and time, indicating that a TCP/IP socket is present, when transmission has been paused.

Additional information

Depending on the current state of your job, you may see additional information displayed to keep you informed about the progress and status of your job. If there is no additional information, you will see (None) displayed.

Mirror status

- **Calculating**—The amount of data to be mirrored is being calculated.
- **In Progress**—Data is currently being mirrored.
- **Waiting**—Mirroring is complete, but data is still being written to the target.

- **Idle**—Data is not being mirrored.
- **Paused**—Mirroring has been paused.
- **Stopped**—Mirroring has been stopped.
- **Removing Orphans**—Orphan files on the target are being removed or deleted depending on the configuration.
- **Verifying**—Data is being verified between the source and target.
- **Unknown**—The console cannot determine the status.

Mirror percent complete

The percentage of the mirror that has been completed

Mirror remaining

The total number of mirror bytes that are remaining to be sent from the source to the target.

Mirror skipped

The total number of bytes that have been skipped when performing a difference. These bytes are skipped because the data is not different on the source and target.

Replication status

- **Replicating**—Data is being replicated to the target.
- **Ready**—There is no data to replicate.
- **Pending**—Replication is pending.
- **Stopped**—Replication has been stopped.
- **Out of Memory**—Replication memory has been exhausted.
- **Failed**—The Double-Take service is not receiving replication operations from the Carbonite Availability driver. Check the Event Viewer for driver related issues.
- **Unknown**—The console cannot determine the status.

Replication queue

The total number of replication bytes in the source queue

Disk queue

The amount of disk space being used to queue data on the source

Bytes sent

The total number of mirror and replication bytes that have been transmitted to the target

Bytes sent compressed

The total number of compressed mirror and replication bytes that have been transmitted to the target. If compression is disabled, this statistic will be the same as **Bytes sent**.

Recovery point latency

The length of time replication is behind on the target compared to the source. This is the time period of replication data that would be lost if a failure were to occur at the current time. This value represents replication data only and does not include mirroring data. If you are mirroring and failover, the data on the target will be at least as far behind as the replication point latency. It could potentially be further behind depending on the circumstances of the mirror. If mirroring is idle and you failover, the data will only be as far behind as the replication point latency time.

Mirror start time

The UTC time when mirroring started

Mirror end time

The UTC time when mirroring ended

Total time for last mirror

The length of time it took to complete the last mirror process

Validating a full server to ESX job

Over time, you may want to confirm that any changes in your network or environment have not impacted your Carbonite Availability job. Use these instructions to validate an existing job.

1. From the **Jobs** page, highlight the job and click **View Job Details** in the toolbar.
2. In the **Tasks** area on the right on the **View Job Details** page, click **Validate job properties**.
3. Carbonite Availability validates that your source and target are compatible. The **Summary** page displays your options and validation items.

Errors are designated by a white X inside a red circle. Warnings are designated by a black exclamation point (!) inside a yellow triangle. A successful validation is designated by a white checkmark inside a green circle. You can sort the list by the icon to see errors, warnings, or successful validations together. Click on any of the validation items to see details. You must correct any errors before you can continue. Depending on the error, you may be able to click **Fix** or **Fix All** and let Carbonite Availability correct the problem for you. For those errors that Carbonite Availability cannot correct automatically, you will need to modify the source or target to correct the error, or you can select a different target. You must revalidate the selected servers, by clicking **Recheck**, until the validation check passes without errors.

4. Once your servers have passed validation, click **Close**.

Editing a full server to ESX job

Use these instructions to edit a full server to ESX appliance job.

1. From the **Jobs** page, highlight the job and click **View Job Details** in the toolbar.
2. In the **Tasks** area on the right on the **View Job Details** page, click **Edit job properties**.
3. You will see the same options for your full server to ESX job as when you created the job, but you will not be able to edit all of them. If desired, edit those options that are configurable for an existing job. See *Creating a full server to ESX job* on page 229 for details on each job option.



Changing some options may require Carbonite Availability to automatically disconnect, reconnect, and remirror the job.

-
4. If you want to modify the workload items or replication rules for the job, click **Edit workload or replication rules**. Modify the **Workload item** you are protecting, if desired. Additionally, you can modify the specific **Replication Rules** for your job.

Volumes and folders with a green highlight are included completely. Volumes and folders highlighted in light yellow are included partially, with individual files or folders included. If there is no highlight, no part of the volume or folder is included. To modify the items selected, highlight a volume, folder, or file and click **Add Rule**. Specify if you want to **Include** or **Exclude** the item. Also, specify if you want the rule to be recursive, which indicates the rule should automatically be applied to the subdirectories of the specified path. If you do not select **Recursive**, the rule will not be applied to subdirectories.

If you need to remove a rule, highlight it in the list at the bottom and click **Remove Rule**. Be careful when removing rules. Carbonite Availability may create multiple rules when you are adding directories. For example, if you add /home/admin to be included in protection, then /home will be excluded. If you remove the /home exclusion rule, then the /home/admin rule will be removed also.

Click **OK** to return to the **Edit Job Properties** page.



If you remove data from your workload and that data has already been sent to the target, you will need to manually remove that data from the target. Because the data you removed is no longer included in the replication rules, Carbonite Availability orphan file detection cannot remove the data for you. Therefore, you have to remove it manually.

-
5. Click **Next** to continue.
 6. Carbonite Availability validates that your source and target are compatible. The **Summary** page displays your options and validation items.

Errors are designated by a white X inside a red circle. Warnings are designated by a black exclamation point (!) inside a yellow triangle. A successful validation is designated by a white checkmark inside a green circle. You can sort the list by the icon to see errors, warnings, or successful validations together. Click on any of the validation items to see details. You must correct any errors before you can continue. Depending on the error, you may be able to click **Fix** or **Fix All** and let Carbonite Availability correct the problem for you. For those errors that

Carbonite Availability cannot correct automatically, you will need to modify the source or target to correct the error, or you can select a different target. You must revalidate the selected servers, by clicking **Recheck**, until the validation check passes without errors.

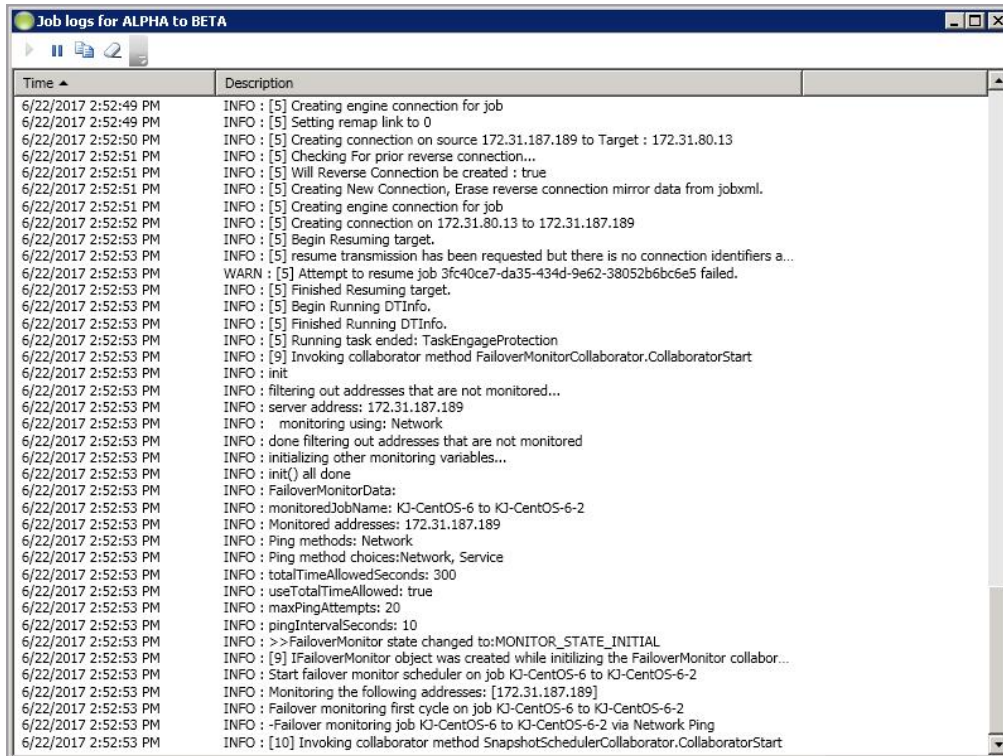
If you receive a path transformation error during job validation indicating a volume does not exist on the target server, even though there is no corresponding data being protected on the source, you will need to manually modify your replication rules. Go back to the **Choose Data** page and under the **Replication Rules**, locate the volume from the error message. Remove any rules associated with that volume. Complete the rest of the workflow and the validation should pass.

After a job is created, the results of the validation checks are logged to the job log. See the *Carbonite Availability and Carbonite Move Reference Guide* for details on the various Carbonite Availability log files.

7. Once your servers have passed validation and you are ready to update your job, click **Finish**.

Viewing a full server to ESX job log

You can view a job log file through the Carbonite Replication Console by selecting **View Job Log** from the toolbar on the **Jobs** page. Separate logging windows allow you to continue working in the Carbonite Replication Console while monitoring log messages. You can open multiple logging windows for multiple jobs. When the Carbonite Replication Console is closed, all logging windows will automatically close.



Time	Description
6/22/2017 2:52:49 PM	INFO : [5] Creating engine connection for job
6/22/2017 2:52:49 PM	INFO : [5] Setting remap link to 0
6/22/2017 2:52:50 PM	INFO : [5] Creating connection on source 172.31.187.189 to Target : 172.31.80.13
6/22/2017 2:52:51 PM	INFO : [5] Checking For prior reverse connection...
6/22/2017 2:52:51 PM	INFO : [5] Will Reverse Connection be created : true
6/22/2017 2:52:51 PM	INFO : [5] Creating New Connection, Erase reverse connection mirror data from jobxml.
6/22/2017 2:52:51 PM	INFO : [5] Creating engine connection for job
6/22/2017 2:52:52 PM	INFO : [5] Creating connection on 172.31.80.13 to 172.31.187.189
6/22/2017 2:52:53 PM	INFO : [5] Begin Resuming target.
6/22/2017 2:52:53 PM	INFO : [5] resume transmission has been requested but there is no connection identifiers a...
6/22/2017 2:52:53 PM	WARN : [5] Attempt to resume job 3fc40ce7-da35-434d-9e62-38052b6bc6e5 failed.
6/22/2017 2:52:53 PM	INFO : [5] Finished Resuming target.
6/22/2017 2:52:53 PM	INFO : [5] Begin Running DTInfo.
6/22/2017 2:52:53 PM	INFO : [5] Finished Running DTInfo.
6/22/2017 2:52:53 PM	INFO : [5] Running task ended: TaskEngageProtection
6/22/2017 2:52:53 PM	INFO : [9] Invoking collaborator method FailoverMonitorCollaborator.CollaboratorStart
6/22/2017 2:52:53 PM	INFO : init
6/22/2017 2:52:53 PM	INFO : filtering out addresses that are not monitored...
6/22/2017 2:52:53 PM	INFO : server address: 172.31.187.189
6/22/2017 2:52:53 PM	INFO : monitoring using: Network
6/22/2017 2:52:53 PM	INFO : done filtering out addresses that are not monitored
6/22/2017 2:52:53 PM	INFO : initializing other monitoring variables...
6/22/2017 2:52:53 PM	INFO : init() all done
6/22/2017 2:52:53 PM	INFO : FailoverMonitorData:
6/22/2017 2:52:53 PM	INFO : monitoredJobName: KJ-CentOS-6 to KJ-CentOS-6-2
6/22/2017 2:52:53 PM	INFO : Monitored addresses: 172.31.187.189
6/22/2017 2:52:53 PM	INFO : Ping methods: Network
6/22/2017 2:52:53 PM	INFO : Ping method choices:Network, Service
6/22/2017 2:52:53 PM	INFO : totalTimeAllowedSeconds: 300
6/22/2017 2:52:53 PM	INFO : useTotalTimeAllowed: true
6/22/2017 2:52:53 PM	INFO : maxPingAttempts: 20
6/22/2017 2:52:53 PM	INFO : pingIntervalSeconds: 10
6/22/2017 2:52:53 PM	INFO : >>FailoverMonitor state changed to:MONITOR_STATE_INITIAL
6/22/2017 2:52:53 PM	INFO : [9] IFailoverMonitor object was created while initializing the FailoverMonitor collabor...
6/22/2017 2:52:53 PM	INFO : Start failover monitor scheduler on job KJ-CentOS-6 to KJ-CentOS-6-2
6/22/2017 2:52:53 PM	INFO : Monitoring the following addresses: [172.31.187.189]
6/22/2017 2:52:53 PM	INFO : Failover monitoring first cycle on Job KJ-CentOS-6 to KJ-CentOS-6-2
6/22/2017 2:52:53 PM	INFO : -Failover monitoring job KJ-CentOS-6 to KJ-CentOS-6-2 via Network Ping
6/22/2017 2:52:53 PM	INFO : [10] Invoking collaborator method SnapshotSchedulerCollaborator.CollaboratorStart

The following table identifies the controls and the table columns in the **Job logs** window.

Start 

This button starts the addition and scrolling of new messages in the window.

Pause 

This button pauses the addition and scrolling of new messages in the window. This is only for the **Job logs** window. The messages are still logged to their respective files on the server.

Copy 

This button copies the messages selected in the **Job logs** window to the Windows clipboard.

Clear

This button clears the **Job logs** window. The messages are not cleared from the respective files on the server. If you want to view all of the messages again, close and reopen the **Job logs** window.

Time

This column in the table indicates the date and time when the message was logged.

Description

This column in the table displays the actual message that was logged.

Failing over full server to ESX jobs

You will be notified in the console when a failover condition has been met. At this time, you should trigger failover. You can also trigger failover at any other time you desire, thus allowing you to better control the failover process.

1. On the **Jobs** page, highlight the job that you want to failover and click **Failover, Cutover, or Recover** in the toolbar.
2. Select the type of failover to perform.
 - **Failover to live data**—Select this option to initiate a full, live failover using the current data on the target. This option will shutdown the source machine (if it is online), stop the protection job, and start the replica virtual machine on the target with full network connectivity.
 - **Perform test failover**—Select this option to perform a test failover.



Test failover is not supported for Btrfs file systems.

In order to use test failover, you must make sure your target has sufficient free disk space available. The free space on each volume group on the target must be larger than 50% of the total size of all logical volumes in that volume group on the source. Meeting that amount of free space may depend on the **Disk Configuration Strategy** you selected under **Replica Virtual Disk Volumes**.

- **Create disks matching source**—With the match source option, you must have sufficient free space on the source before the job is created because the target disks will be matching the source. You may need to increase free disk space on the source (perhaps add a partition which has been created on the raw disk on the source and then extend the volume group), in order to allow for sufficient free space to be matched on the target.
 - **Create disks per volume**—With the per volume option, select a volume group size on the target that has enough free space to accommodate the 50% free space requirement.
-
- The source, target, and protection job will remain online and uninterrupted during the test.
 - The test will be performed using the test failover settings configured during job creation.
 - The test will use the current data on the target.
 - Scheduled snapshots will be deferred during the test and taken automatically after the test is undone.
 - The test failover will take a snapshot of the current data on the target and create a new set of virtual disks. The data from the snapshot will be mirrored to the new set of disks using the same mirroring options as the protection job.
 - Once the mirror is complete, the replica virtual machine or alternate replica virtual machine, depending on your selected configuration, is automatically brought online using the new set of disks.

- The replica virtual machine or alternate replica virtual machine, depending on your selected configuration, will use the network settings specified in the test failover settings of the protection job.
 - When you are finished with your test, undo it.
 - When you undo a test failover, the new set of disks will be maintained or deleted as specified in the test failover settings of the protection job.
 - At any time during a test failover, you can undo the test, perform a live failover, or failover to a snapshot. (Performing a live failover or failing over to a snapshot will automatically undo any test in progress.)
- **Failover to a snapshot**—Select this option to initiate a full, live failover without using the current data on the target. Instead, select a snapshot and the data on the target will be reverted to that snapshot. This option will not be available if there are no snapshots on the target. To help you understand what snapshots are available, the **Type** indicates the kind of snapshot.
 - **Scheduled**—This snapshot was taken as part of a periodic snapshot.
 - **Deferred**—This snapshot was taken as part of a periodic snapshot, although it did not occur at the specified interval because the job between the source and target was not in a good state.
 - **Manual**—This snapshot was taken manually by a user.
3. Select how you want to handle the data in the target queue.
 - **Apply data in target queues before failover or cutover**—All of the data in the target queue will be applied before failover begins. The advantage to this option is that all of the data that the target has received will be applied before failover begins. The disadvantage to this option is depending on the amount of data in queue, the amount of time to apply all of the data could be lengthy.
 - **Discard data in the target queues and failover or cutover immediately**—All of the data in the target queue will be discarded and failover will begin immediately. The advantage to this option is that failover will occur immediately. The disadvantage is that any data in the target queue will be lost.
 4. When you are ready to begin failover, click **Failover**.
 5. If you performed a test failover, you can undo it by selecting **Undo Failover or Cutover** in the toolbar. The replica virtual machine on the target will be shut down and deleted if you chose to use an alternate virtual machine for the test. The virtual disks for used for the test will be deleted.
-



If you need to update DNS after failover, there is a sample DNS update script located in `/etc/DT/sysprep.d`. You may need to modify the script for your environment. If you need basic assistance with script modifications, contact technical support. Assistance with advanced scripting will be referred to Professional Services.

There is no reverse or fallback once you have failed over.

Reversing protection after failover for full server to ESX jobs

There is no automated reverse or failback for a full server to ESX appliance job once you have failed over. If you need to go back to your original hardware, you will need to create a new job in the opposite direction following one of the processes below, depending on the original source.

- **Physical server**—Use these steps if your original source is a physical server.
 1. Resolve the problems on the original source that caused it to fail. If you need to deploy a new server, use the same operating system and disk configuration as the original source.
 2. If Carbonite Availability is still running on the original source, replace the license since that license is currently running on the failed over server. If Carbonite Availability is no longer installed, reinstall it with an appropriate license.
 3. Create a full server job from the failed over server to your original source. See *Creating a full server job* on page 184 for details on creating this job.
 4. Once the initial mirror is complete, failover the full server job. See *Failing over full server jobs* on page 219 for details on this process.
- **ESX virtual server**—Use these steps if your original source is a virtual server on an ESX host.
 1. Delete the original source virtual server from the ESX host. If you want to reuse the .vmdk files again, only delete the original source virtual server from the ESX inventory.
 2. If you do not have one already, create a virtual recovery appliance on the ESX host where the original source virtual server is located. This appliance will be the target of the new job you are going to create. This appliance needs Carbonite Availability installed and licensed on it. For more details, see *Full server to ESX requirements* on page 223.
 3. Create a full server to ESX job from your failed over server to the appliance on the ESX host where the original source virtual server is located. See *Creating a full server to ESX job* on page 229 for details on creating this job.
 4. Once the initial mirror is complete, failover the full server to ESX job. See *Failing over full server to ESX jobs* on page 272 for details on this process.
- **Hyper-V virtual server**—Use these steps if your original source is a virtual server on a Hyper-V host.
 1. Resolve the problems on the original source that caused it to fail. If you need to deploy a new server, use the same operating system and disk configuration as the original source.
 2. If Carbonite Availability is still running on the original source, replace the license since that license is currently running on the failed over server. If Carbonite Availability is no longer installed, reinstall it with an appropriate license.
 3. Create a full server job from the failed over server to your original source. See *Creating a full server job* on page 184 for details on creating this job.
 4. Once the initial mirror is complete, failover the full server job. See *Failing over full server jobs* on page 219 for details on this process.

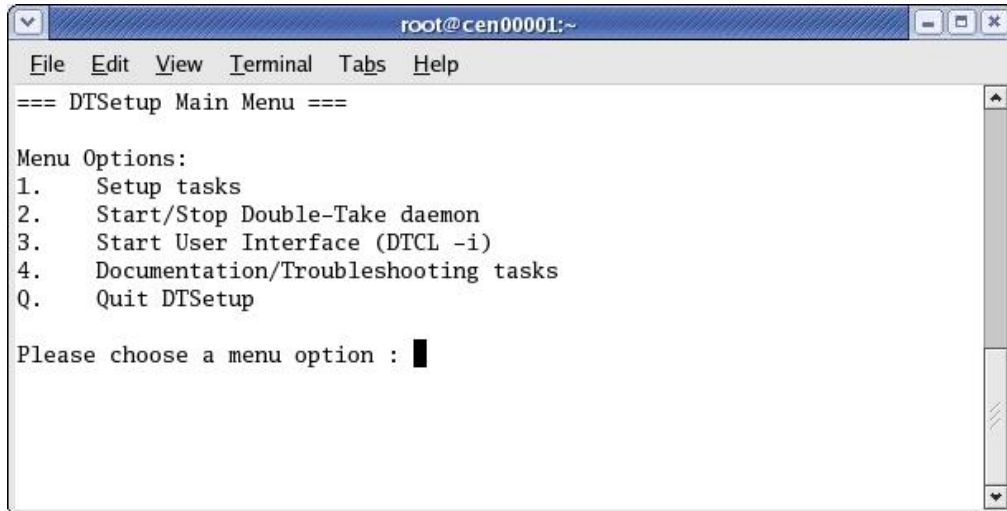
Chapter 6 DTSetup

DTSetup is a menu-driven application that provides easy access to Carbonite Availability server configuration. Select a link for more information on DTSetup configuration tasks.

- *Running DTSetup* on page 276—This topic includes instructions for launching DTSetup.
- *Setup tasks* on page 277—The setup tasks allow you to configure license keys, security groups, block device replication configuration, server configuration, and driver performance settings.
- *Starting and stopping the service* on page 282—Built-in scripts allow you to quickly and easily start and stop the Carbonite Availability service.
- *Starting DTCL* on page 283—You can launch the Carbonite Availability interactive command prompt which allows you to enter DTCL commands one at a time.
- *Viewing documentation and troubleshooting tools* on page 284—DTSetup provides easy access to Carbonite Availability log files, a diagnostic collection tool, and several legal documents.
- *DTSetup menus* on page 285—This topic includes a list overview of the DTSetup menu system. Reference the links in the list for complete details on completing tasks in DTSetup.

Running DTSetup

1. Run the DTSetup command from the shell prompt to start DTSetup. The command is case-sensitive.
2. The first time you run DTSetup after an installation, you will be prompted to review the Carbonite license agreement. Review the agreement and accept the terms of agreement by typing yes. You cannot use Carbonite Availability without agreeing to the licensing terms.
3. When the DTSetup menu appears, enter the number of the menu option you want to access.

A screenshot of a terminal window titled "root@cen00001:~". The window contains the following text:

```
File Edit View Terminal Tabs Help
=== DTSetup Main Menu ===

Menu Options:
1. Setup tasks
2. Start/Stop Double-Take daemon
3. Start User Interface (DTCL -i)
4. Documentation/Troubleshooting tasks
Q. Quit DTSetup

Please choose a menu option : █
```

Setup tasks

The setup tasks are generally configured once. Select a link below to learn more about that setup task.

- *Activating your server* on page 278—License keys and activation keys license and activate your Carbonite Availability servers.
- *Modifying security groups* on page 279—Security groups provide access to Carbonite Availability.
- *Configuring server settings* on page 280—If desired, you can modify server settings through the Carbonite Availability configuration file.
- *Configuring driver performance settings* on page 281—If desired, you can specify Carbonite Availability driver performance settings.

Activating your server

Before you can use Carbonite Availability, each source and target server must have a valid license key, which is an alpha-numeric codes that applies the appropriate Carbonite Availability license to your installation.

1. Start DTSetup. See *Running DTSetup* on page 276.
2. Select **Setup tasks**.
3. Select **Set License Key Menu**.
4. Select **Set License Key in /etc/DT/DT.conf**.
5. Enter your license key and press Enter. The license key will automatically be inserted into the configuration file. You are prompted to start the Carbonite Availability service after the first installation, and you must restart the service each time the license key is modified, such as after an upgrade.
6. Press Enter to return to the menu.
7. Press Q as many times as needed to return back to the main menu or to exit DTSetup.

Modifying security groups

During the installation, the user root is automatically added to the Carbonite Availability administrators security group. If you want to add other users or remove root, you will need to modify the security group configuration for each source and target server. See *Security* on page 286 for more details on the security groups and the privileges granted to each group.

1. Start DTSetup. See *Running DTSetup* on page 276.
2. Select **Setup tasks**.
3. Select **Add/Remove users to Double-Take groups**.
4. Select the necessary menu options to add or remove groups to the administrator or monitors group as needed, and specify the user name when prompted.
5. When you have completed your security group modifications, press Q as many times as needed to return back to the main menu or to exit DTSetup.

Configuring server settings

Server settings are available in various places. You can access them via the Replication Console for Linux, through DTCL, or through DTSetup. Initially, the server settings file, `/etc/DT/DT.conf`, on the source and target is blank. To populate it with default values, start and stop the Double-Take service once.

1. Start DTSetup. See *Running DTSetup* on page 276.
2. Select **Setup tasks**.
3. Select **Edit Double-Take config file**.
4. The server settings are listed in alphabetical order. Make modifications as necessary, using the control keys specified at the bottom of the page. For a complete list of each server setting, valid values, default values, and optional notes, see *Server Settings* in the *Scripting Guide*.
5. Press control-X to exit the configuration file.
6. Enter Yes or No to save any changes.
7. Press Q as many times as needed to return back to the main menu or to exit DTSetup.

Configuring driver performance settings

Driver settings provide configuration flexibility so you can adjust Carbonite Availability based on your servers, network, and replication requirements. You may want to modify driver settings on both the source and target.



Changing the driver performance settings can have a positive or negative impact on server performance. These settings are for advanced users. If you are uncertain how to best modify the driver performance settings, contact technical support.

1. Start DTSetup. See *Running DTSetup* on page 276.
2. Select **Setup tasks**.
3. Select **Configure Double-Take driver performance**.
4. The current driver settings are displayed.
5. Select a driver setting to modify the option.
 - **Toggle Adaptive Throttling**—You can toggle between enabling (true) and disabling (false) **Adaptive Throttling**. This occurs when kernel memory usage exceeds the **Throttling Start Level** percentage. When throttling is enabled, operations are delayed by, at most, the amount of time set in **Maximum Throttling Delay**, thus reducing kernel memory usage. Throttling stops when the kernel memory usage drops below the **Throttling Stop Level** percentage.
 - **Toggle Forced Adaptive Throttling**—You can toggle between enabling (true) and disabling (false) **Forced Adaptive Throttling**. This causes all operations to be delayed by, at most, the amount of time in set in **Maximum Throttling Delay**, regardless of the kernel memory being used. **Adaptive Throttling** must be enabled (true) in order for **Forced Adaptive Throttling** to work.
 - **Set Maximum Throttling Delay**—This option is the maximum time delay, in milliseconds, used by the driver during a system delay.
 - **Set Throttling Delay Interval**—This option is the interval, in milliseconds, to check memory usage during a throttling delay. If a delay is no longer needed, the remainder of the delay is skipped.
 - **Set Throttling Start Level**—Throttling starts when disk writes reach the specified percentage. This prevents the driver from stopping replication because memory has been exhausted.
 - **Set Throttling Stop Level**—Throttling stops when disk writes reach the specified percentage.
 - **Set Memory Usage Limit**—This option is the amount of kernel memory, in bytes, used for queuing replication operations. When this limit is exceeded, the driver will send an error to the service forcing a remirror of all active connections.
 - **Set Maximum Write Buffer Size**—This option is the maximum amount of system memory, in bytes, allowed for a single write operation. Operations exceeding this amount are split into separate operations in the queue.
6. After you have completed your driver performance modifications, press Q as many times as needed to return back to the main menu or to exit DTSetup.

Starting and stopping the service

The Double-Take service will start automatically after Carbonite Availability is installed and the server is rebooted. You can start and stop the Double-Take service using this built-in DTSetup script.

1. Start DTSetup. See *Running DTSetup* on page 276.
2. Select **Start/Stop Double-Take service**.
3. Select the necessary menu option to start or stop the service and handle the driver configuration.
 - **Start Double-Take and process driver config**—This option starts the Double-Take service and loads the Carbonite Availability drivers.
 - **Stop Double-Take but preserve driver config**—This option stops the Double-Take service but does not unload the Carbonite Availability drivers.
 - **Restart service but preserve driver config**—This option does a full stop and start of the Double-Take service but does not unload the Carbonite Availability drivers.
 - **Restart service and reset driver config**—This option does a full stop and start, completely unloading the Double-Take service and Carbonite Availability drivers and then reloading them.
 - **Stop the running service and teardown driver config**—This option stops the Double-Take service and the Carbonite Availability drivers are unloaded.
 - **Go to Replication Configuration menu**—This option takes you to **Setup Tasks, Configure Block Device Replication**. When you press Q to exit from that menu, you will return this menu.
4. When you have completed your starting and stopping tasks, press Q as many times as needed to return back to the main menu or to exit DTSetup.

Starting DTCL

You can launch the Carbonite Availability interactive command prompt which allows you to enter DTCL commands one at a time.

1. Start DTSetup. See *Running DTSetup* on page 276.
2. Select **Start User Interface (DTCL -i)**.
3. Enter your DTCL commands one at a time at the **Command** prompt. For a complete list of DTCL commands, their syntax, and instructions for completing tasks using DTCL, see the *Scripting Guide*.
4. To exit the DTCL **Command** prompt, type exit.
5. When you have completed your DTCL tasks, press Q as many times as needed to return back to the main menu or to exit DTSetup.

Viewing documentation and troubleshooting tools

1. Start DTSetup. See *Running DTSetup* on page 276.
2. Select **Documentation/Troubleshooting tasks**.
3. Select **View log files** to view the following log files. Carbonite Availability logs alerts, which are processing notifications, warnings, and error messages. The logs are written to disk.
 - **View /*.dtl in less**—This option uses the less file viewer program to view all of the Carbonite Availability logs, starting from the most recent.
 - **Follow the output of latest**—This option uses tail -f to watch the output of the Carbonite Availability logs in real-time.
 - **View /var/log/messages in less**—This option uses the less file viewer program to view the system log messages.
 - **Follow the output of /var/log/messages**—This option uses tail -f to watch the output of the system log messages in real-time.
4. Select one of the **Collect and package diagnostic info** selections to run the DTInfo script which collects configuration data. This can be useful when reporting problems to technical support. Depending on the diagnostic option you select, the amount of data to be collected varies between basic, detailed and full diagnostic information. You must have root (or uid 0 equivalent) to execute the diagnostics or to copy or read the resulting file.
5. Select **View user documentation** to view several legal documents. DTSetup attempts to determine your viewers, although you can specify your viewer.
 - **View End User License Agreement TXT**—This option views the End User License Agreement legal document.
 - **View driver module license TXT**—This option views the open source legal document.
 - **Change a document viewer**—This option allows you to specify a document viewer.
6. When you have completed your documentation and troubleshooting tasks, press Q as many times as needed to return back to the main menu or to exit DTSetup.

DTSetup menus

The following lists is an overview of the DTSetup menu system. Reference the links for complete details on completing tasks in DTSetup.

1. **Setup tasks**—License keys, security groups, replication configuration, server configuration, and driver performance settings. See *Setup tasks* on page 277.
 1. **Set License Key Menu**—See *Activating your server* on page 278.
 2. **Add/Remove users to Double-Take groups**—See *Modifying security groups* on page 279.
 3. **Edit Double-Take config file**—See *Configuring server settings* on page 280.
 4. **Configure Double-Take driver performance**—See *Configuring driver performance settings* on page 281.
2. **Start/Stop Double-Take service**—See *Starting and stopping the service* on page 282.
3. **Start User Interface (DTCL -i)**—See *Starting DTCL* on page 283.
4. **Documentation/Troubleshooting tasks**—See *Viewing documentation and troubleshooting tools* on page 284.

Chapter 7 Security

To ensure protection of your data, Carbonite Availability offers multi-level security using native operating system security features. Privileges are granted through membership in user groups defined on each machine. To gain access to a source or target, the user must provide a valid local user account that is a member of one of the Carbonite Availability security groups. Once a valid user name and password have been provided and the source or target has verified membership in one of the security groups, the user is granted appropriate access to the source or target and the corresponding features are enabled in the client. Access is granted on one of the following three levels.

- **Administrator Access**—All features are available for that machine.
- **Monitor Access**—Servers and statistics can be viewed, but functionality is not available.
- **No Access**—Servers appear in the clients, but no access to view the server details is available.

Although passwords are encrypted when they are stored, Carbonite security design does assume that any machine running the client application is protected from unauthorized access. If you are running the client and step away from your machine, you must protect your machine from unauthorized access.

Adding users to the security groups

The security groups are automatically created during the installation process. The groups can be local or LDAP (Lightweight Directory Access Protocol). The groups are called dtadmin (default group ID 501) and dtmon (default group ID 502). During the installation, the user root is automatically added to the dtadmin group.

Users that need administrator access to Carbonite Availability must be added to the dtadmin group. Users that need monitor only access must be added to the dtmon group. In both cases, you must provide a valid local user account.

1. Run the DTSetup command from the shell prompt. The command is case-sensitive.
2. Select **Setup tasks**.
3. Select **Add/Remove users to Double-Take groups**.
4. Select the necessary menu options to add or remove groups to the administrator or monitors group as needed, and specify the user name when prompted.
5. When you have completed your security group modifications, press Q as many times as needed to return back to the main menu or to exit DTSetup.

Chapter 8 Special network configurations

Carbonite Availability can be implemented with very little configuration necessary in small or simple networks, but additional configuration may be required in large or complex environments. Because an infinite number of network configurations and environments exist, it is difficult to identify all of the possible configurations. Review the following sections for configuration information for that particular type of network environment.

- *Firewalls* on page 289
- *IP and port forwarding* on page 290

Firewalls

If your source and target are on opposite sides of a firewall, you will need to configure your hardware to accommodate communications. You must have the hardware already in place and know how to configure the hardware ports. If you do not, see the reference manual for your hardware.

- **Carbonite Availability ports**—Ports 1500, 1505, 1506, 6325, and 6326 are used for Carbonite Availability communications and must be open on your firewall. Open UDP and TCP for both inbound and outbound traffic.
- **ESX ports**—If you are using VirtualCenter or an ESX host, port 443 is also required and must be opened.

You need to configure your hardware so that the Carbonite Availability ports and ESX ports applicable to your environment are open. Since communication occurs bidirectionally, make sure you configure both incoming and outgoing traffic.

There are many types of hardware on the market, and each can be configured differently. See your hardware reference manual for instructions on setting up your particular router.

IP and port forwarding

As outlined in the requirements, Carbonite Availability supports IP and port forwarding in NAT environments with the following caveats.

- Only IPv4 is supported.
- Only standalone servers are supported.
- DNS failover and updates will depend on your configuration
 - Only the source or target can be behind a router, not both.
 - The DNS server must be routable from the target

When setting up a job in an environment with IP or port forwarding, make sure you specify the following configurations.

- Make sure you have added your server to the Carbonite Replication Console using the correct public or private IP address. The name or IP address you use to add a server to the console is dependent on where you are running the console. Specify the private IP address of any servers on the same side of the router as the console. Specify the public IP address of any servers on the other side of the router as the console. This option is on the **Add Servers** page in the **Manual Entry** tab.

Add Servers

Identify the servers in your environment that you want to manage. The servers you add here appear on the Servers page.

Manual Entry | Automatic Discovery

Server: 112.47.12.7

User name: domain\administrator

Password:

Domain:

Management Service port: 1025 Use default port

Add

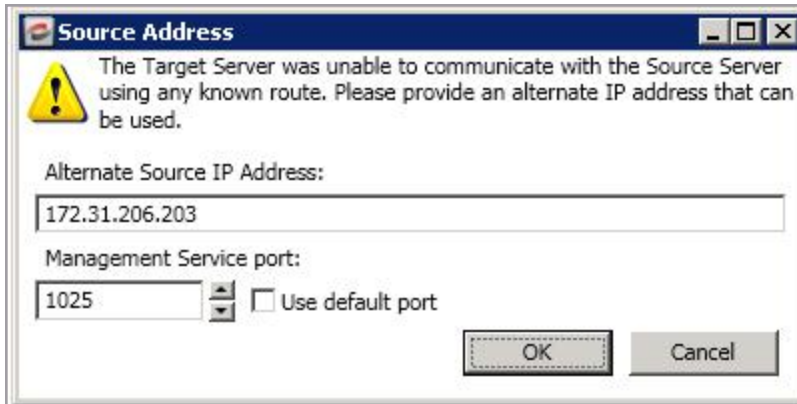
Servers to be added:

Server	Details
--------	---------

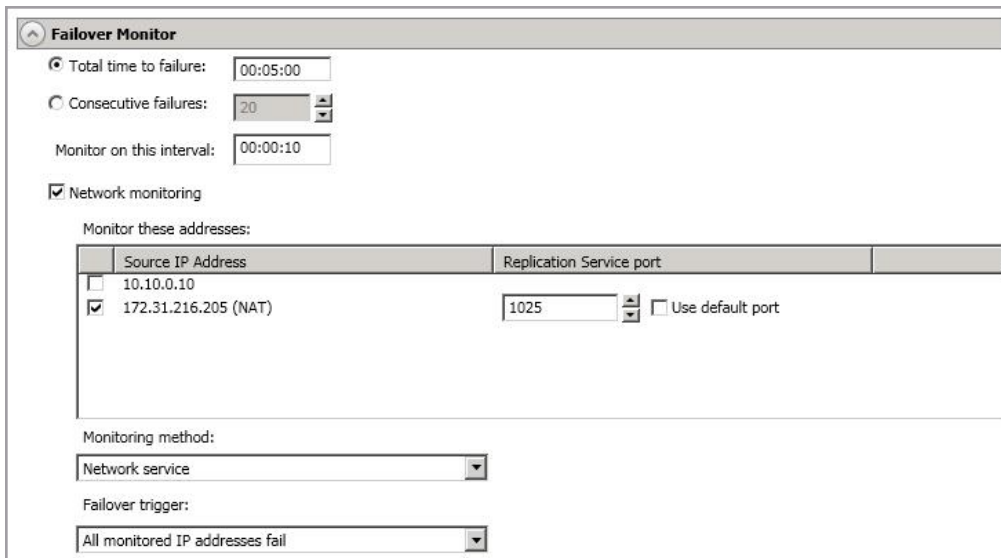
Remove Remove All

OK Cancel

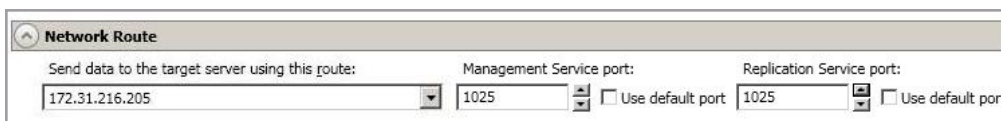
- When choosing the target server for your job, you may be prompted for a route from the target to the source. This route, and a port if you are using a non-default port, is used so the target can communicate with the source to build job options. This dialog box will be displayed, only if needed, after you click **Next** on the **Choose Target** page in the job creation wizard.



- If you are configuring network monitoring, make sure you specify the port to use for monitoring the Double-Take replication service. This option is in the job creation wizard on the **Set Options** page in the **Failover Monitor** section.



- When specifying a network route or reverse routes for full server jobs, you can enter a public IP address and then specify ports for the Double-Take Management Service and Double-Take replication service. This option is in the job creation wizard on the **Set Options** page in the **Network Route** section or the **Reverse Protection and Routing** section for full server jobs.



Reverse Protection and Routing

Send data to the target server using this route: Management Service port: Use default port Replication Service port: Use default port

Enable reverse protection

A reserved IP address permanently identifies each server so that failover and reverse can both be performed. The reserved IP addresses will not be moved on failover or reverse. These addresses will also be used to route the data in non-NAT environments.

Select a reserved IP address on the source:

Send data to source after reverse using this route: Replication Service port: Use default port

Select a reserved IP address on the target: