CARBONITE

Carbonite Availability and Carbonite Move

Reporting Service User's Guide

Notices

Carbonite Availability and Carbonite Move Reporting Service User's Guide Version 8.1.1, Friday, February 23, 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.

Contents

Carbonite Availability and Carbonite Move Reporting Service overview

The Carbonite Availability and Carbonite Move Reporting Service is a centralized reporting and analysis tool that allows you to create detailed, custom reports of all Carbonite Availability and Carbonite Move servers in your environment. It can be used to show the overall effectiveness of Carbonite Availability and Carbonite Move protection over time and to analyze trends in a data protection scheme. After you install and configure Carbonite Availability and Carbonite Move Reporting Service, it monitors and collects data from the Carbonite Availability and Carbonite Move servers you specify. It sends the collected data to a SQL database that you create and maintain.

If you want to use Reporting Service, you will need to complete the following steps, in order.

1. Create your SQL database according to the requirements. See *Reporting Service requirements* on page 5 for details on requirements for your environment and your SQL documentation for detailed instructions on configuring SQL server and databases.
2. Install Carbonite Availability and Carbonite Move Reporting Service.
3. Using the Carbonite Replication Console from another machine, add the server where you installed Reporting Service to your console. See *Adding a Reporting Service server to the Carbonite Replication Console* on page 7.
4. Using the Carbonite Replication Console, configure the Reporting Service server. See *Configuring the Reporting Service server* on page 8.

Once your configuration is complete, Carbonite Availability and Carbonite Move data will be collected and you can use standard SQL queries and tools to create your own customized reports. See your SQL documentation for detailed instructions on creating queries and reports. For a complete list of the data stored in the Reporting Service tables, see *Reporting Service tables* on page 12.

Reporting Service requirements

To use Carbonite Availability and Carbonite Move Reporting Service, your environment must meet the following requirements.

- The reporting service must be installed on a Windows operating system.
- The reporting service cannot be installed on a Carbonite Availability and Carbonite Move source or target server.
- You will need the Carbonite Replication Console installed on a Windows machine in order to configure Reporting Service. The console can be on your Carbonite Availability and Carbonite Move source or target server, or it can be on a Carbonite Availability and Carbonite Move client-only machine. The reporting service can be run with a Carbonite Availability and Carbonite Move client-only installation.
- You must have an existing server running SQL Server 2005 or later.
- The reporting service can be run on the SQL server or a different server.
- A new SQL database needs to be created, however no tables should be created because Carbonite Availability and Carbonite Move Reporting Service will automatically create the tables it needs.
- The newly created SQL database needs established security credentials. The database user role membership must initially include db_owner and public to allow Reporting Service to create the tables it needs. Once the tables have been created, the database user role membership can be changed, if desired, to db_datareader, db_datawriter, and public.

  You need to use Management Studio to set your security credentials. Under **Security**, **Logins**, right-click on the SQL user (the SQL user, not a Windows user) and open the properties. In the Login Properties window, select **User Mapping** to see a list of databases and role memberships for the selected database. Select **Map** for the database and select the database roles. See your SQL documentation for complete details.

- Your SQL server authentication must be set to SQL Server and Windows Authentication mode.
- The following networking requirements apply. See your SQL documentation for details on these settings.
  - Your SQL server must have TCP/IP enabled.
  - The SQL server must be set to listen on all IP addresses.
  - All IP addresses on the SQL server must be using the same port, however the addresses do not have to be using the default port. If you are not using the default port, you must change the following configuration file to specify the port you are using.
    1. Remove the read-only attribute from \Program Files\Carbonite\Replication\Reporting Service\reporting-service.properties.
    2. Modify the following line, changing 1433 to the SQL port you are using.

       ```
       visionsolutions.managementservice.dbport=1433
       ```

    3. Reapply the read-only attribute to the file.
    4. Stop and restart the Reporting Service.

5. Verify that the database configuration in the Carbonite Replication Console successfully connects.

- If you are using a firewall on your SQL server, make sure it does not block SQL or Reporting Service traffic.

Adding a Reporting Service server to the Carbonite Replication Console

You will need the Carbonite Replication Console installed on a Windows machine in order to configure Reporting Service. The console can be on your Carbonite Availability and Carbonite Move source or target server, or it can be on a Carbonite Availability and Carbonite Move client-only machine. The reporting service can be run with a Carbonite Availability and Carbonite Move client-only installation.

See the Carbonite Replication Console help for complete details on the console.

1. You have two methods to get to the **Add Servers** page in the console.
   - Click **Get Started** from the toolbar, select **Add servers**, and then click **Next**.
   - Click **Servers** from the toolbar and click **Add Servers** from the second toolbar on the **Servers** page.
2. On the **Manual Entry** tab, specify the server information.
   - **Server**—This is the name or IP address of the Reporting Service server.
   - **User name**—Specify a user that is a member of the **Double-Take Admin** 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**.
3. After you have specified the server information, click **Add**.
4. Click **OK** to add the server to your console.

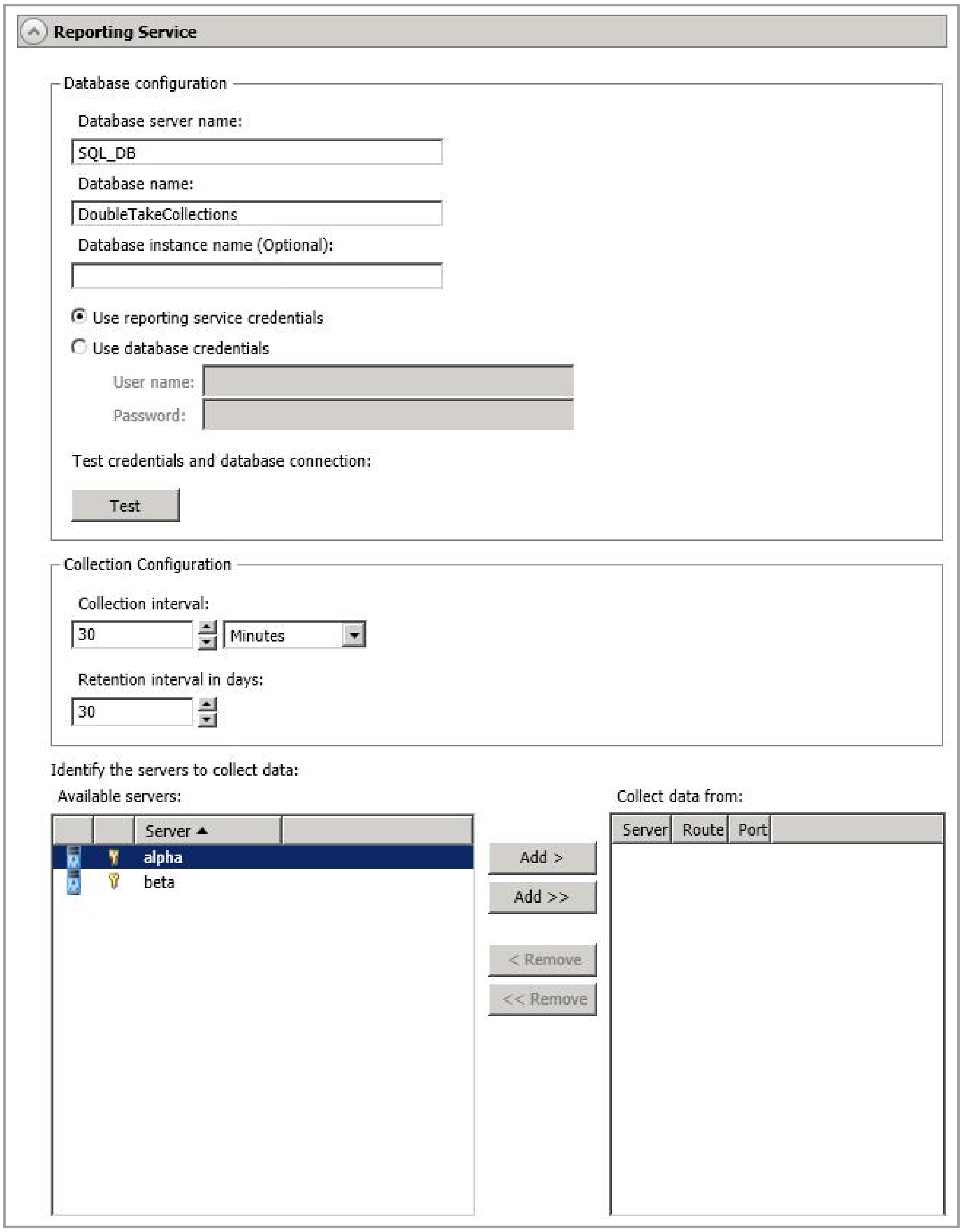
Configuring the Reporting Service server

1. Make sure you have inserted your Reporting Service server into your console. See the console help for details on adding a server to the console.
2. From the **Servers** page, double-click your Reporting Service server to open the server's details.
3. From the **View Collector Details** page, click the **Edit server properties** link.
4. The Reporting Service properties identify the SQL database configuration, the data collection configuration, and the Carbonite Availability and Carbonite Move servers you are collecting from.

   - **Database server name**—Specify the name of the SQL server that contains your SQL database that will store the collected Carbonite Availability and Carbonite Move data.
   - **Database name**—Specify the name of the SQL database that will store the collected Carbonite Availability and Carbonite Move data.
   - **Database instance name**—If necessary, specify the database instance name of the SQL database that you specified.
   - **Use reporting service credentials**—Select this option if you want to use the credentials the Reporting Service is running as.
   - **Use database credentials**—Select this option if you want to specify SQL database credentials.
   - **Test**—This button will test the specified credentials and check to see if the tables exist in the specified database. If they do not exist, they will be created. The database user role membership must be db_owner and public to create the tables. If the tables already exist, they will be updated to the correct version, if necessary. Once the test (and therefore the table creation or verification) is complete, the database user role membership can be changed, if desired, to db_datareader, db_datawriter, and public.
   - **Collection interval**—Specify the amount of time to wait between data collections.
   - **Retention interval in days**—Specify how long to retain the collected data. Data older than the specified number of days will be deleted from the database.
   - **Identify the servers to collect data**—Only the servers in your console session will be listed. Highlight the servers you want to collect data from and click **Add >**. If you want to add all of the servers click **Add >>**. If the server you want to collect data from is not listed, you need to add it from the **Servers** page.

     If desired, specify the **Route** and **Port** to use for communication to the server you are collecting data from.

     If you need to remove a server from the **Collect data from** list, click **< Remove**. If you want to remove all of the servers, click **<< Remove**.



Reporting Service

Database configuration

Database server name: SQL_DB

Database name: DoubleTakeCollections

Database instance name (Optional):

● Use reporting service credentials

○ Use database credentials

User name:

Password:

Test credentials and database connection:

Collection Configuration

Collection interval: 30 Minutes

Retention interval in days: 30

Identify the servers to collect data:

Available servers:

Server
alpha
beta

Collect data from:

Server	Route	Port

5. When you have finished your Reporting Service server configuration, click **OK** to return to the **Servers** page.

Viewing Reporting Service server details

The **View Collector Details** page allows you to view details about a Reporting Service server.

Server name

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

Roles

The role of this server in your Carbonite Availability and Carbonite Move 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 communcations. You will also see the Carbonite Availability and Carbonite Move protocol being used to communicate with server. The protocol will be XML web services protocol (for servers running Carbonite Availability and Carbonite Move 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

Database server

The name of the server that contains your SQL database that storing your collected Carbonite Availability and Carbonite Move data

Database name

The name of the SQL database that is storing your collected Carbonite Availability and Carbonite Move data

Database instance

The optional database instance name

Database status

The status of the database configuration

Collection interval

The amount of time Reporting Service waits between data collections

Retention interval

The length of time to retain collected data. Data older than the specified number of days will be deleted from the database

Server

The server Reporting Service is collecting Carbonite Availability and Carbonite Move data from

Route

The route the reporting service is using to communicate with the server data is being collected from

Port

The port the reporting service is using to communicate with the server data is being collected from

User

The user the reporting service is using to authenticate with the server data is being collected from

Status

The status of the collection of data from the server

Reporting Service tables

The following data is available in the Reporting Service tables. Each time data is collected, it is written to a new row in the table. You can link data together using time stamps and unique IDs to see how data and states change over time.

Server table

ServerGUID

The unique ID Carbonite Availability and Carbonite Move assigns to the server

Timestamp

The date and time the server data was collected

Name

The name of the server

Status

The status of the server

Job table

JobId

The unique ID that Carbonite Availability and Carbonite Move assigns to the job

Timestamp

The date and time the job data was collected

SourceUniqueID

The unique ID Carbonite Availability and Carbonite Move assigns to the job's source server

TargetUniqueID

The unique ID Carbonite Availability and Carbonite Move assigns to the job's target server

SourceHostUri

The source's URI (uniform resource identifier)

TargetHostUri

The target's URI (uniform resource identifier)

Name

The name that Carbonite Availability and Carbonite Move assigned to the job. You may have modified the job name.

Workload

The type of workload the job is using

Type

The job type. This value equates to the **Job Type** column in the top right pane on the **Jobs** page. See the managing and controlling jobs topic for any job type for additional details.

CreatorUserName

The name of the user that created the job

Health

The high level health, or state, of the job. This value equates to the colored icons seen in the first column in the top right pane on the **Jobs** page. See the managing and controlling jobs topic for any job type for additional details.

HighLevelState

The overall state of the job. This value equates to the **Activity** column in the top right pane on the **Jobs** page. See the managing and controlling jobs topic for any job type for additional details.

LowLevelState

The low level state of the job. This value equates to the **Additional Information** field in the bottom right pane on the **Jobs** page. See the managing and controlling jobs topic for any job type for additional details.

TargetState

The state of the data on the target. This value equates to the **Target data state** field in the bottom right pane on the **Jobs** page. See the managing and controlling jobs topic for any job type for additional details.

CanEdit

Indicates if the job can currently be edited

CanDelete

Indicates if the job can currently be deleted

CanStart

Indicates if the job can currently be started

CanStop

Indicates if the job can currently be stopped

CanPause

Indicates if the job can currently be paused

CanFailover

Indicates if the job can currently be failed over

CanFailback

Indicates if the job can currently be failed back

CanRestore

Indicates if the job can currently be restored

CanReverse

Indicates if the job can currently be reversed

CanUndoFailover

Indicates if a failed over job can currently be undone

Connection table

ManagedConnectionId

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.

JobId

The unique job ID associated with this connection

Timestamp

The date and time the connection data was collected

SourceUniqueID

The unique ID Carbonite Availability and Carbonite Move assigns to the job's source server

TargetUniqueID

The unique ID Carbonite Availability and Carbonite Move assigns to the job's target server

BandwidthCollar

The bandwidth limiting that has been set or the keyword **Unlimited** if no bandwidth limit has been set

CompressionEnabled

Indicates if data is being compressed before it is sent to the target

CompressionLevel

The level of compression

DiskQueueBytes

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

InitialMirrorComplete

Indicates if the initial mirror has been completed

MirrorBytesRemaining

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

MirrorBytesSent

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

MirrorBytesSkipped

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.

MirrorBytesTransmitted

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

MirrorOpsQueued

The total number of mirror operations in the queue

MirrorPermillage

The percentage of the mirror that has been completed

MirrorState

The state of mirroring. This value equates to the **Mirror Status** column in the top right pane on the **Jobs** page. See the managing and controlling jobs topic for any job type for additional details.

PeerMemoryLow

Indicates if the target is running low on memory based on the **Amount of system memory to use** setting on the target server's queue properties. See the console help for details on Carbonite Availability and Carbonite Move queue settings.

ReplicationBytesQueued

The total number of replication bytes in the source queue

ReplicationBytesSent

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

ReplicationBytesTransmitted

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

ReplicationOpsQueued

The total number of replication operations in the queue

ReplicationState

The state of replication. This value equates to the **Replication Status** column in the top right pane on the **Jobs** page. See the managing and controlling jobs topic for any job type for additional details.

Restoring

Identifies if the connection is actively restoring

SourceAvailable

Identifies if the target was able to communicate with the source server

SourceEngineAvailable

Identifies if the target was able to communicate with Carbonite Availability and Carbonite Move on the source

SourceMachineName

The name of the server associated with this connection

StartTime

The date and time the connection was initiated

TargetAvailable

Identifies if the source was able to communicate with the target server

TargetEngineAvailable

Identifies if the source was able to communicate with Carbonite Availability and Carbonite Move on the target

TargetRoute

The IP address identifying the route to the target

TargetMachineName

The name of the target server associated with this connection

TargetQueueBytes

The number of bytes queued on the target

TargetState

The state of the target

TotalBytesSent

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

TotalBytesTransmitted

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

TotalOpsQueued

The total number of mirror and replication operations that are in the source queue

TransmissionMode

Indicates if data is actively being transmitted to the target

SourceClusterResourceState

The state of the Double-Take Source Connection resource, if it is being used by a cluster-aware job on a Carbonite Availability and Carbonite Move source cluster

CurrentTime

The date and time the connection data was written to the database

MirrorStartTime

The data and time when the mirroring process started

MirrorEndTime

The data and time when the mirroring process ended

RecoveryPointTime

The date and time replication is synchronized between the source and target. The difference between the TimeStamp and this time is the time period of replication data that would be lost if a failure were to occur at the TimeStamp. 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 recovery point time. 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 recovery point time.

RecoveryPointLatency

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.

Version table

This table is used internally by Reporting Service for versioning. Do not modify or use this table.