Symantec™ Disaster Recovery Orchestrator Administration Guide

Microsoft Azure

6.1

May 2014

Symantec

Symantec™ Disaster Recovery Orchestrator Administration Guide

The software described in this book is furnished under a license agreement and may be used only in accordance with the terms of the agreement.

Product version: 6.1

Document version: 6.1 Rev 0

Legal Notice

Copyright © 2014 Symantec Corporation. All rights reserved.

Symantec, the Symantec Logo, the Checkmark Logo, Veritas, Veritas Storage Foundation, CommandCentral, NetBackup, Enterprise Vault, and LiveUpdate are trademarks or registered trademarks of Symantec Corporation or its affiliates in the U.S. and other countries. Other names may be trademarks of their respective owners.

The product described in this document is distributed under licenses restricting its use, copying, distribution, and decompilation/reverse engineering. No part of this document may be reproduced in any form by any means without prior written authorization of Symantec Corporation and its licensors, if any.

THE DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID. SYMANTEC CORPORATION SHALL NOT BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS DOCUMENTATION. THE INFORMATION CONTAINED IN THIS DOCUMENTATION IS SUBJECT TO CHANGE WITHOUT NOTICE.

The Licensed Software and Documentation are deemed to be commercial computer software as defined in FAR 12.212 and subject to restricted rights as defined in FAR Section 52.227-19 "Commercial Computer Software - Restricted Rights" and DFARS 227.7202, "Rights in Commercial Computer Software or Commercial Computer Software Documentation", as applicable, and any successor regulations, whether delivered by Symantec as on premises or hosted services. Any use, modification, reproduction release, performance, display or disclosure of the Licensed Software and Documentation by the U.S. Government shall be solely in accordance with the terms of this Agreement.

Symantec Corporation
350 Ellis Street
Mountain View, CA 94043

http://www.symantec.com

Technical Support

Symantec Technical Support maintains support centers globally. Technical Support's primary role is to respond to specific queries about product features and functionality. The Technical Support group also creates content for our online Knowledge Base. The Technical Support group works collaboratively with the other functional areas within Symantec to answer your questions in a timely fashion. For example, the Technical Support group works with Product Engineering and Symantec Security Response to provide alerting services and virus definition updates.

Symantec's support offerings include the following:

- A range of support options that give you the flexibility to select the right amount of service for any size organization
- Telephone and/or Web-based support that provides rapid response and up-to-the-minute information
- Upgrade assurance that delivers software upgrades
- Global support purchased on a regional business hours or 24 hours a day, 7 days a week basis
- Premium service offerings that include Account Management Services

For information about Symantec's support offerings, you can visit our website at the following URL:

www.symantec.com/business/support/index.jsp

All support services will be delivered in accordance with your support agreement and the then-current enterprise technical support policy.

Contacting Technical Support

Customers with a current support agreement may access Technical Support information at the following URL:

www.symantec.com/business/support/contact_techsupp_static.jsp

Before contacting Technical Support, make sure you have satisfied the system requirements that are listed in your product documentation. Also, you should be at the computer on which the problem occurred, in case it is necessary to replicate the problem.

When you contact Technical Support, please have the following information available:

- Product release level
- Hardware information

- Available memory, disk space, and NIC information
- Operating system
- Version and patch level
- Network topology
- Router, gateway, and IP address information
- Problem description:
  - Error messages and log files
  - Troubleshooting that was performed before contacting Symantec
  - Recent software configuration changes and network changes

Licensing and registration

If your Symantec product requires registration or a license key, access our technical support Web page at the following URL:

www.symantec.com/business/support/

Customer service

Customer service information is available at the following URL:

www.symantec.com/business/support/

Customer Service is available to assist with non-technical questions, such as the following types of issues:

- Questions regarding product licensing or serialization
- Product registration updates, such as address or name changes
- General product information (features, language availability, local dealers)
- Latest information about product updates and upgrades
- Information about upgrade assurance and support contracts
- Information about the Symantec Buying Programs
- Advice about Symantec's technical support options
- Nontechnical presales questions
- Issues that are related to CD-ROMs or manuals

Support agreement resources

If you want to contact Symantec regarding an existing support agreement, please contact the support agreement administration team for your region as follows:

Asia-Pacific and Japan	customercare_apac@symantec.com
Europe, Middle-East, and Africa	semea@symantec.com
North America and Latin America	supportsolutions@symantec.com

About Symantec Connect

Symantec Connect is the peer-to-peer technical community site for Symantec's enterprise customers. Participants can connect and share information with other product users, including creating forum posts, articles, videos, downloads, blogs and suggesting ideas, as well as interact with Symantec product teams and Technical Support. Content is rated by the community, and members receive reward points for their contributions.

http://www.symantec.com/connect/storage-management

Documentation

Your feedback on product documentation is important to us. Send suggestions for improvements and reports on errors or omissions. Include the title and document version (located on the second page), and chapter and section titles of the text on which you are reporting. Send feedback to:

doc_feedback@symantec.com

For information regarding the latest HOWTO articles, documentation updates, or to ask a question regarding product documentation, visit the Storage and Clustering Documentation forum on Symantec Connect.

https://www-secure.symantec.com/connect/storage-management/forums/storage-and-clustering-documentation

Contents

Chapter 1

Introduction to Disaster Recovery Orchestrator

This chapter includes the following topics:

About Symantec Disaster Recovery Orchestrator for Microsoft Azure

Symantec Disaster Recovery Orchestrator provides the following services for applications running on an organization's on-premises systems and in the Microsoft Azure cloud environment:

- Application monitoring
- Disaster recovery (DR)

Application monitoring

Disaster Recovery Orchestrator provides monitoring capabilities for the applications running in an organization's IT environment. The on-premises systems on which these applications are deployed can be physical computers or virtual machines that

are managed in a virtualization environment. In a cloud environment, the applications are deployed on virtual machines. Any virtualization platform may be used.

Disaster Recovery Orchestrator employs an agent framework to monitor the state of applications and their dependent components. Specific agents are available to monitor the application, storage, and network components. Together, these agents monitor the overall health of the configured applications by running specific commands, tests, or scripts.

Disaster Recovery Orchestrator is based on Symantec Cluster Server, and uses similar concepts such as agents, resources, and service groups to provide application monitoring. Disaster Recovery Orchestrator has a lightweight server footprint that allows faster installation and configuration.

Disaster recovery

When the Disaster Recovery Orchestrator solution is in place, the Azure cloud acts as a DR site for the applications that run at the on-premises site. In the event of a disaster, you can continue servicing your clients through the applications running in Azure. Later, you can resume application processing at the on-premises site when it is functional again. Even when the on-premises site is fully operational, you can perform a fire drill to test the DR readiness of the configuration.

Application monitoring and disaster recovery workflow

The Disaster Recovery Orchestrator architecture uses an agent framework to monitor the state of the applications and their dependent components running on a system. Disaster Recovery Orchestrator agents monitor the overall health of the configured applications by running specific commands, tests, or scripts.

Application monitoring workflow

When you configure application monitoring, Disaster Recovery Orchestrator Client begins to monitor the application components. If an application or a component fails, the Client agents attempt to restart it for a configurable number of times. If all the restart attempts fail, the status of the application is set to Faulted.

Disaster recovery workflow

The high-level disaster recovery (DR) operations need to be performed manually.

To configure an application for DR and to protect from a failure, perform the following operations:

1. Configure an on-premises application for DR.

The Applications page of Disaster Recovery Orchestrator Console begins to display the status of its heartbeat, availability, and data replication.

2. Check the DR-readiness of your configuration.

   Perform the fire drill operation on a dedicated Azure virtual network. This operation does not affect any operations on the on-premises site.

3. Take over the application processing and monitoring at the Azure site. You may do so as part of a planned maintenance event or to recover from a disaster at the on-premises site.

   If an on-premises application stops responding, the Applications page displays the change in its status. Perform a takeover operation on the application.

4. When the on-premises site is functional again, resume processing and monitoring the application as normal.

   Perform a failback operation on the application.

About Disaster Recovery Orchestrator agents

Agents are modules that plug into the Disaster Recovery Orchestrator framework, and that help manage the components of the configured applications and the various infrastructure resources.

The agents are installed when you install Disaster Recovery Orchestrator Client. These agents start, stop, and monitor the components of the configured applications and report their state changes. If an application or its components fail, these agents restart the applications and their components on the system.

A system requires one agent per component to monitor all the components of that type. For example, a single GenericService agent manages all the services that are configured using the GenericService components. When the agent starts, it obtains the necessary configuration information from these components and then monitors the configured applications. The agent then periodically updates Disaster Recovery Orchestrator with the component and application status.

Agents perform the following operations:

- Brings the components online
- Takes the components offline
- Monitors the components and reports the state changes

Disaster Recovery Orchestrator agents are classified as follows:

- Infrastructure agents
  These agents are packaged with the base software, and they include agents for mount points, network cards and ports, generic services, heartbeats, and

processes. These agents are immediately available for use after you install Disaster Recovery Orchestrator.

- Application agents
  These agents are used to monitor third-party applications such as Microsoft SQL Server, custom applications, and so on. For further information about the Disaster Recovery Orchestrator agent for a supported application, refer to the corresponding configuration guide.

About resource monitoring

Disaster Recovery Orchestrator employs an event-based monitoring framework to determine the status of the configured application and its components. This framework is called the Intelligent Monitoring Framework (IMF), and it is implemented using custom as well as native operating system-based notification mechanisms.

IMF provides instantaneous state change notifications. Disaster Recovery Orchestrator agents detect this state change and then trigger the necessary actions.

IMF provides the following key benefits:

- Instantaneous notification
  Faster fault detection results in faster failover and thus less application down time.
- Reduction in system resource utilization
  Conventional resource monitoring occurs every 60 seconds by default. With IMF event-based monitoring there is less reliance on conventional monitoring and so this interval can be increased. Thus Disaster Recovery Orchestrator reduces CPU utilization and provides significant benefits in terms of system resource utilization.
- Ability to monitor large number of components
  Due to the ability to increase conventional monitor cycle intervals, IMF allows monitoring of more components with a lower system resource utilization.

How IMF works

IMF is enabled by default for a component if its Disaster Recovery Orchestrator agent supports IMF.

The following steps outline how IMF-based monitoring works:

1. A Disaster Recovery Orchestrator agent waits for the components to report the same steady state (either Online or Offline) for two consecutive monitor cycles. Then, it registers the components for IMF-based monitoring.

2. The agent then registers itself for receiving certain operating system-specific or custom event notifications.
3. If a component fails, the agent executes a monitor cycle to determine its state. If the state is Offline, Disaster Recovery Orchestrator takes the necessary corrective action, depending on the configuration.
4. If the component state remains the same, the agent moves to a Wait state and then waits for the next event to occur.

Criteria for the applications that Disaster Recovery Orchestrator can monitor

Most applications can be placed under Disaster Recovery Orchestrator control provided the following guidelines are met:

- Defined start, stop, and monitor procedures exist.
  The application to be monitored must have defined procedures for starting, stopping, and monitoring, as follows:

Start procedure	The application must have a command to start it and all the dependent components and resources that it may require. Disaster Recovery Orchestrator brings up the required resources in a specific order and then brings up the application using the defined start procedure.
Stop procedure	The application must have a command to stop it and all its dependent components and resources. Disaster Recovery Orchestrator stops the application using the defined stop procedure, and then stops the required resources in the reverse order in which they were started.
Monitor procedure	The application must have a monitor procedure that determines if the specified application instance is healthy. The application must allow individual monitoring of unique instances. For example, in a database environment, the monitoring application can perform SQL commands to verify read and write access to the database. The closer a test comes to matching what a user does, the better the test is in discovering problems. You should balance the level of monitoring between ensuring that the application is up and minimizing the monitor overhead.

- Ability to restart the application in a known state
  When the application is stopped, it must close out all tasks, store data properly, and then exit. When Disaster Recovery Orchestrator attempts to restart the

application, it should be able to start from the last known state. In case of a server crash, the application must be able to recover gracefully.

Commercial databases such as SQL Server and Oracle are examples of crash-tolerant applications. On any client request, the client is responsible for holding the request until it receives acknowledgment from the server. When the server receives a request, it is placed in a special redo log file. The database confirms that the data is saved before it sends an acknowledgment to the client.

After a server crashes, the database recovers to the last-known committed state by mounting the data tables and applying the redo logs. This recovery returns the database to the time of the crash. The client resubmits any outstanding client requests that are unacknowledged by the server, and all others are contained in the redo logs.

Common terms used in the context of Disaster Recovery Orchestrator

The Disaster Recovery Orchestrator solution caters to on-premises and cloud environments, and therefore, deals with a wide range of entities. Some of these entities may be referred to using multiple names. This section describes some common terms and conventions that are used throughout the Disaster Recovery Orchestrator user interface and documentation. These terms are listed in an alphabetical order.

Application Monitoring

A feature of Disaster Recovery Orchestrator that enables you to monitor applications running on a physical computer or a virtual machine. If the application components fail and cannot be recovered after a certain number of attempts, the application status is reported accordingly on the Health View or the Console UI.

Application Monitoring Configuration

An application that is configured for monitoring on the on-premises application host or the cloud application host. Disaster Recovery Orchestrator Client manages the application monitoring configuration on a system.

Application Recovery Configuration

An application that is configured for migration or for disaster recovery (DR) in the cloud in the event of a failure at the on-premises site. Disaster Recovery Orchestrator Console manages all the application recovery configurations. An application must be configured for monitoring before it can be configured for DR.

The following systems are associated with every application recovery configuration:

- On-premises application host
- Console host
- Cloud application host

These systems are also referred to as the systems that participate in the DR solution for an application.

Application Virtual Machine or Cloud Application Host

The Azure virtual machine on which Disaster Recovery Orchestrator Client is installed. This virtual machine acts as the Azure counterpart of on-premises system that hosts the application that is configured for recovery. The application is configured for monitoring on this virtual machine as well. If the on-premises application or its host becomes unavailable a recovery administrator performs Takeover. When Takeover is successful, this virtual machine begins processing the application.

Client

The Disaster Recovery Orchestrator client component that manages the authentication, file replication, application monitoring, and user interface modules.

Client Host

The on-premises system or the Azure virtual machine on which Disaster Recovery Orchestrator Client is installed. This system hosts the application and its monitoring configuration.

Cloud

The Microsoft Azure cloud service platform.

For more information, see the Microsoft article:

http://azure.microsoft.com/en-us/overview/what-is-azure/

Console

The Disaster Recovery Orchestrator server component that manages the authentication, file replication, DR, and user interface modules.

Console Host

The Azure virtual machine on which Disaster Recovery Orchestrator Console is installed. This virtual machine acts as controller for the DR activities.

Console UI

The Disaster Recovery Orchestrator Console user interface, which is browser-based.

Failback

The operation in which application processing is restored on the original on-premises system when it becomes available again. A recovery administrator manually triggers Failback, but the tasks involved in the operation are performed automatically in a predefined sequence.

File Replication

The replication mechanism that Disaster Recovery Orchestrator uses to synchronize application data between the on-premises site and the Azure site.

Firedrill

A feature of Disaster Recovery Orchestrator that lets you test your DR configuration. A fire drill operation tests the Takeover operations on an application configured for DR. When a fire drill is successful, the application comes online in a separate Azure virtual network, without disrupting the application in the production environment.

Journal file

The intermediate file that is used to store information about the updates made to the application data folders at the primary site. This information is further used to replicate those updates at the secondary site. This file is also referred to as the replication log.

On-Premises System or On-Premises Application Host

A physical computer or virtual machine that exists on the premises of an organization, rather than in the cloud. This system hosts the application that is configured for monitoring and then further configured for recovery.

Primary

The system or location that is the source for data replication, where data synchronization is required for recovery in the event of an operational failure. For example, while the application processes requests from the on-premises application host, that system is the primary and the Console host is the secondary.

Recovery or Disaster Recovery (DR)

A feature of Disaster Recovery Orchestrator that enables you to recover application processing in the cloud when your organization's on-premises site becomes unavailable. You can restore application processing back to the on-premises site when it is available again.

You can also perform a planned migration of the application from the on-premises site to the cloud site.

Recovery Administrator

A user who has the privileges to configure an application for recovery, perform recovery operations, and remove the application recovery configuration. A security administrator adds this user to the Privilege Settings tab of the Console UI.

Replication Log Volume

A dedicated volume on the storage that is attached to the systems that participate in the DR solution. This volume is used to store the journal file.

Secondary

The system or location that is the destination for data replication, where data synchronization is required for recovery in the event of an operational failure. For example, while the application processes requests from the cloud application host, that virtual machine is the primary and the on-premises application host is the secondary.

Security Administrator

A user who has the privileges to configure the recovery settings and other users for Disaster Recovery Orchestrator Console. The security administrator cannot directly work with application recovery configurations.

Takeover

The operation in which application processing is taken over by the cloud application host, when your on-premises application or its host or the site becomes unavailable. A recovery administrator manually triggers Takeover, but the tasks involved in the operation are performed automatically in a predefined sequence.

Virtualization Host

A host software that creates and manages virtual machines, for example, Microsoft Hyper-V Server.

Virtual Machine

A software-based computer that is provisioned to run certain processes or provide some specific services, like hosting an application.

Chapter 2

Configuring applications for recovery

This chapter includes the following topics:

About configuring applications for disaster recovery

To recover an application at the cloud site when the on-premises site becomes unavailable, you must configure the application for monitoring and disaster recovery (DR).

After you configure an application for monitoring using the Health View, you must further configure it for disaster recovery (DR). This enables you to recover the application processing at the cloud site when the on-premises site becomes unavailable.

The Disaster Recovery Orchestrator Configuration Wizard is used to configure an application for DR, and it performs the following tasks:

- Configures the application monitoring service on the Client hosts (the on-premises system and the corresponding cloud virtual machine that host the application and the Disaster Recovery Orchestrator Client components).

- Configures single sign-on (SSO) with the Client hosts.
- Configures the file replication service and the privileges associated with it on the Console host.
  The wizard adds DCOM privileges for the file replication service to the systems that are associated with an appliction recovery configuration. The replication service is used to synchronize data between the systems that host the application.
- Configures file replication monitoring and infrastructure monitoring on both the Client hosts.
- Updates the Console database with this configuration information.
- Finalizes the application recovery configuration.

When finalizing the application recovery configuration, the wizard performs the following tasks:

- Takes the application configuration offline on the cloud application host.
- Deprovisions the virtual machine.
- Attaches the data disks to the Console host.
- Changes the replication settings on the Console host.
- Starts data replication from the on-premises application host.
- Brings the application configuration online on the on-premises application host.

Signing in to Disaster Recovery Orchestrator Console

To access the Disaster Recovery Orchestrator Console UI, you use a web browser.

To sign in to Disaster Recovery Orchestrator Console, you must have one of the following privileges:

- Disaster Recovery Orchestrator security administrator
  The first user that logs in to the UI is the security administrator. This user defines other Disaster Recovery Orchestrator Console users and their privileges.
- Recovery administrator for an application
  The security administrator defines this user.
- Guest user for an application
  The security administrator defines this user.

Refer to the *Symantec Disaster Recovery Orchestrator Deployment Guide* for information about the following requirements for working with Disaster Recovery Orchestrator:

- Users and privileges
- Web browser settings

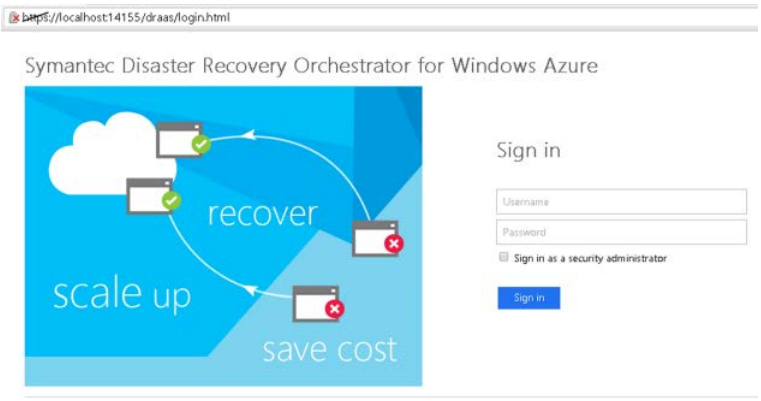
To sign in to the Console

1 Open a supported web browser, and enter the following URL:

`https://ConsoleHost:14155/draas/login.html`

Replace the *ConsoleHost* variable with the name of the virtual machine that hosts the Console, or its IP address. On the Console host itself, you may replace *ConsoleHost* with **localhost**.

The following figure depicts the login page:



2 Enter a valid user name.

- To sign in as a security administrator:
  - If you are not a domain user, enter only your user name.
  - If you are a domain user, enter the domain name and your user name in the format `Domain\UserName`.
  - Select the **Sign in as a security administrator** check box.

  Note: The user signing in as the security administrator must have local administrator privileges on the Console host.

- To sign in as a recovery administrator or a guest user, provide only the user name.

  Do not select the check box.

3 Enter the password corresponding to the user name entered previously.

4 Click **Sign in**.

- When you sign in as a security administrator, you can manage the recovery settings and the privilege settings for Disaster Recovery Orchestrator Console.
  See "About recovery settings" on page 23.
  See "Managing recovery administrators and guest users" on page 24.
- When you sign in as a recovery administrator, you can configure an on-premises application for recovery in the cloud, view its status, and perform operations on it.
  See "Configuring an application for disaster recovery" on page 27.
  See "About administering disaster recovery of applications" on page 31.
- When you sign in as a guest, you can view the status of the application recovery configurations and their reports.
  See "Viewing a summary of all the applications recovery configurations" on page 40.
  See "Viewing reports of disaster recovery operations" on page 55.

About the Disaster Recovery Orchestrator Console settings

The Settings view of the Disaster Recovery Orchestrator Console UI allows you to view and change various security settings.

- The Privileges Settings page allows you to define users and their privileges. These users are the recovery administrators who perform various administrative activities on the applications.
  See "Managing recovery administrators and guest users" on page 24.
- The Recovery Settings page allows you to view and change generic settings for Disaster Recovery Orchestrator Console.
  See "About recovery settings" on page 23.

About recovery settings

The Settings view of the Disaster Recovery Orchestrator Console UI contains the Recovery Settings tab, which allows you to view and change the generic recovery settings.

The Recovery Settings page displays the following information:

- Subscription ID assosciated with the current instance of Disaster Recovery Orchestrator Console
- Name of the Disaster Recovery Orchestrator Console service
- Location of the Personal Information Exchange (PFX) file that is used to store the authentication information for a secure cloud service
- Name of the bubble network to be used for performing fire drills on the applications
- The subnet used for the fire drill network

The Recovery Settings page allows you to change the following information:

- PFX file location and password
- Fire drill network and its subnet
- Security administrator (Active Directory user) name and password

If you change the values in the editable fields, click **Confirm** on the command bar to save the changes. If the values that you provided are valid, saves the changes. Otherwise, an error message is displayed. Fix the issue mentioned in the message, and then try to save your changes again.

Managing recovery administrators and guest users

To configure and administer the recovery of applications using Disaster Recovery Orchestrator Console, you need the following kinds of users:

- A user who is designated as the security administrator for Disaster Recovery Orchestrator Console
  This user must be a local administrator on the Console host, but may or may not be a domain user.
  The same user can be designated as a recovery administrator on one or more application virtual machines.
  This user must select the **Sign in as a security administrator** check box on the login page to successfully sign in to the Console UI.
- At least one user corresponding to each application host with the Admin privileges, called the recovery administrator
  Recovery administrators can configure and administer applications for recovery.
  Users with the Guest privileges on the application hosts can only view the corresponding application, operations, and reports.

Only a security administrator for Disaster Recovery Orchestrator can add and manage the recovery administrators and guest users for applications. To do so, the security administrator must sign in to the Console UI and open the **Settings > Privilege Settings** view.

User names and host names are not case-sensitive, but they must be unique.

To add a recovery administrator

1. On the command bar at the bottom, click **Add**.
2. On the Configure Users and Groups dialog box, specify the following:
   - Enter the name of the user who is to be designated as a recovery administrator for an application.
     Enter only the name that was provided when creating the domain user; do not enter the domain in this field. Email addresses are also not allowed.
   - Enter the name of the on-premises system that hosts the application.
   - Select the privilege.
     Admin users can perform operations on the application recovery configuration.
     Guest users can only view the application and its operations and reports.
3. Click the check mark icon at the bottom right to save and close the dialog box.

To modify a recovery administrator

1. Select a row on the Privilege Settings tab.
2. On the command bar, click **Edit**.
3. On the Configure Users and Groups dialog box, change the values in the required fields.
4. Click the check mark icon at the bottom right to save and close the dialog box.

To delete a recovery administrator

1. Select a row on the Privilege Settings tab.

   Note: Check whether an application on the corresponding system has not been configured for recovery. If such an application exists and if you delete a user who has the Admin privileges, any recovery operations cannot be performed on the application. If you accidentally delete such a user, make sure to again add the user with the same privileges for the same system.

2. On the command bar, click **Delete**.
3. On the confirmation dialog that appears, click **Yes**.

   This user can no longer perform operations on, or view the status of, the application on that system.

Considerations for configuring an application for recovery

This section lists the considerations for configuring an application for disaster recovery (DR).

Software and network

Consider the following software configuration requirements before configuring an application for DR:

- Make sure that the latest Adobe Flash Player plugin is available for the browser that you use to access the Console UI.
- Adobe Flash Player must be enabled for use on systems that run Windows Server 2012.
  See "Disaster Recovery Orchestrator Console views are not displayed correctly" on page 106.
- The authentication (AT) service requires that an endpoint with the port number 14153 is configured on the Console host and the Client hosts.
- The application that you want to configure for DR must be configured for monitoring on the on-premises application host and the cloud application host. If application monitoring is not configured on any of these systems, the Disaster Recovery Orchestrator Configuration Wizard prompts you to do so. You can launch the Application Monitoring Configuration Wizard from within the DR configuration wizard. After the application monitoring configurations are in place on both the systems, you can proceed to configure the application for recovery.

Configuration

- Ensure that User Access Control (UAC) is disabled on all the systems that participate in the DR solution.
- Ensure that the appropriate users are configured on the Privilege Settings view of the Console UI.

Storage and replication

Consider the following storage and replication requirements before configuring an application for DR:

- During the time that Disaster Recovery Orchestrator Console is hosted on the virtual machine, a disk must not be manually attached or detached. Disaster Recovery Orchestrator must manage the addition or removal of any storage on this virtual machine.

- The disks must not be part of a storage pool. If you use a volume created in a storage pool, the takeover and failback operations will eventually fail.
- The Windows automount feature must be enabled on the Console host. The replication service driver needs the volumes to be mounted so that it can access the file replication configurations.
  For more information, see the Microsoft article:
  http://technet.microsoft.com/en-us/library/cc753703.aspx
- Sufficient space must be available on the volumes that are used to store the application data.
  If required, the volumes can be resized even after replication is configured.
- The application data must be stored at identical locations on the on-premises application host and the corresponding cloud application host. This is required for configuring replication between the two sites, which ensures that application data is synchronized.

  The following criteria must be satisfied:
  - The folders to be mapped for replication must exist at both locations.
  - The drive letters of the volumes on which the folders are located must match exactly.
- The journal file size must be defined appropriately. Although the minimum requirement is 1 GB, Symantec recommends that you set the journal file size to 10 GB for better performance.
  Specify a size that fits within the space that is currently available on the volume. You can change the journal file size for each application later from the corresponding Settings page.
- The disk on which the journal file is located (replication log volume) must not be detached while the replication is in progress.

Configuring an application for disaster recovery

To configure an application for disaster recovery (DR), it must first be configured for monitoring. The Disaster Recovery Orchestrator Configuration Wizard checks for application monitoring configurations on the on-premises application host and the cloud application host.

To configure an application for disaster recovery

1. Sign in to the Disaster Recovery Orchestrator Console UI from a web browser.
2. On the command bar, click **Configure**.
3. On the On-Premises System Information panel, provide the following input:

- Select the name of the on-premises system that hosts the application.
- Enter the user name and password of a domain user who has the privileges to configure the application for DR.
  You may specify the current user or a different user. However, the user must have local administrator privileges on the on-premises system.

Click the Next arrow.

The wizard searches for application monitoring configurations on the specified system, and proceed as follows:

- If the wizard does not find any application monitoring configurations, it displays a message and prompts you to configure an application for monitoring. Click **Configure** to launch the Application Monitoring Configuration Wizard, and step through the wizard.

  Note: Make sure that pop-up blockers are not enabled on the browser.

  See "Configuring application monitoring" on page 59.
  After you exit the Application Monitoring Configuration Wizard, click the right arrow at the bottomr right corner on the DR configuration wizard.
- If the wizard finds any applications that are configured for monitoring but not configured for DR, it displays the next page.

4 On the Application Virtual Machine Mapping page, specify the following:

- Select the on-premises application that you want to map to a cloud virtual machine for DR.
- Select the name of the cloud virtual machine that hosts the application.
- Enter the user name and password of a domain user who has the privileges to configure the application for DR.
  You may specify the current user or a different user. However, the user must have local administrator privileges on the cloud application host.

Click the Next arrow.

The wizard searches for application monitoring configurations on the specified virtual machine, and proceeds as it did earlier for the on-premises application in step 3. Take the appropriate action to proceed.

5 On the Data Mapping for Replication panel, and map the application data folders to the corresponding folders on the cloud application host.

If the Disaster Recovery Orchestrator Configuration Wizard is able to identify the data folders that configured for the application, they are selected by default.

If you do not want to replicate any specific folders, remove them from the Selected Folders list on the right.

Click the Next arrow.

6 On the Replication Journal Information panel, provide the following information:

- A location and size for the on-premises journal file
- A location and size for the cloud journal file

Click the Next arrow.

7 On the Virtual Computer Name panel, specify the following:

- Select the IP address to be used to access the on-premises application.
- Enter a unique virtual name for the application.
- Provide the credentials of the user in whose context the application monitoring helper service runs.
  You may specify the current user or a different user. However, the user must have DNS administrator privileges and must be a local administrator on the on-premises application host and the corresponding cloud application host.

Note: If the user that you specify does not have the appropriate privileges, the DR configuration might fail.

Click the Next arrow.

8 On the Summary panel, review the data that the wizard has collected so far.

Click the Next arrow.

9 On the Implementation panel, review the progress of the tasks as the wizard performs them.

If an issue occurs, the wizard displays an error message and provides a link to the logs that you can use for troubleshooting.

When all the tasks are completed, click the Next arrow.

10 On the DR Site Preparation panel, take one of the following actions:

- To finalize the application configuration immediately, click **Configure now**.

See “Finalizing the application recovery configuration” on page 30.

- To finalize the application configuration later, click **Configure later**.
  You must remember to perform this operation by clicking the appropriate link from the Applications view later.

If you do not click either of these buttons and exit the wizard, you can finalize the application later.

Finalizing the application recovery configuration

This procedure is the last step in configuring an application for disaster recovery (DR).

You can finalize an application recovery configuration in one of the following ways.

- On the Finalize Application Recovery Configuration panel of the DR configuration wizard, click **Finalize**.
  The wizard prompts you to confirm whether it should proceed with the final tasks.
  - If you click **Yes**, it proceeds with the tasks required to complete application recovery configuration, and displays the status of each task.
    When all the tasks are completed, click the check mark icon in the lower right corner to exit the wizard.
  - If you click **No**, it does not proceed. You will need to complete the final tasks from the Applications view later.
- On the Applications view of Disaster Recovery Orchestrator Console, click the **Finalize application recovery configuration** link. The wizard prompts you to confirm whether it should proceed with the final tasks, and if you click **Yes**, it displays the Configure dialog box.
  While the tasks are in progress, use the Close button on the top right corner to temporarily close the dialog box. You can click the **Finalize application recovery configuration** link again to reopen the dialog box.
  When all the tasks are completed, click the check mark icon in the lower right corner to close the dialog box.

Chapter 3

Administering application recovery

This chapter includes the following topics:

About administering disaster recovery of applications

To administer disaster recovery (DR) of applications, you perform the following high-level operations:

- **Configure**: Configure an on-premises application for monitoring and DR.
  See "About configuring applications for disaster recovery" on page 20.

- **Firedrill**: In preparation for a disaster, you can test the DR configuration on the cloud site even as the primary site is fully operational.
  See "Testing a disaster recovery configuration by performing a fire drill" on page 48.
- **Takeover**: In the event of a disaster or during a planned maintenance event at the on-premises site, continue processing and monitoring the application from the cloud site.
  See "Taking over application processing at the cloud site" on page 49.
- **Failback**: When the on-premises site is functional again, failback the operations that were running in the cloud.
  See "Failing back application processing to the on-premises site" on page 51.
- **Unconfigure**: Stop the monitoring and data replication of a configured on-premises application. The application can no longer be recovered in the cloud environment in case of a failure at the primary site.
  See "Removing the recovery configuration of an application" on page 57.

Additionally, you can perform the following administrative tasks:

- Start or stop the replication activity for an application
  See "Starting or stopping the replication of application data" on page 52.
- Change the settings of an application recovery configuration
  See "Changing the disaster recovery settings of an application" on page 54.
- View the reports of the operations performed on the configured applications
  See "Viewing reports of disaster recovery operations" on page 55.

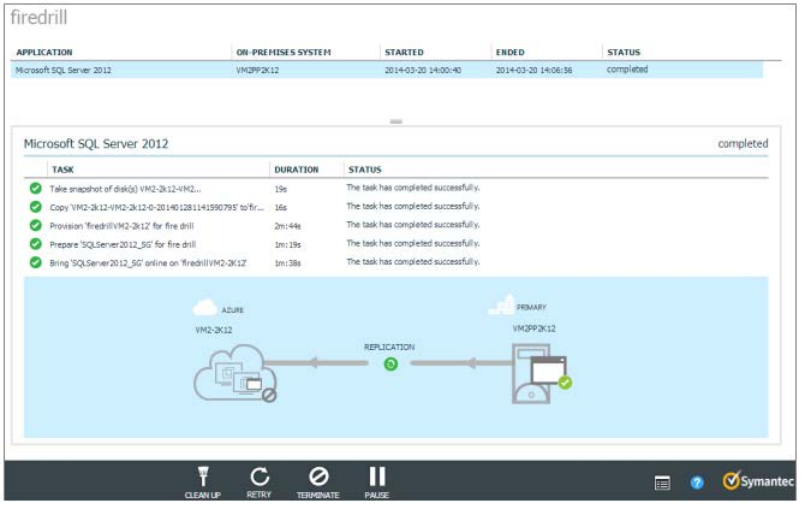
Note the following considerations for administering DR of applications:

- Only a recovery administrator can change the application recovery configuration settings and perform takeover, failback, or fire drill operations on a configured application.
  See "Managing recovery administrators and guest users" on page 24.
- To perform any administrative activity on a configured application, you must sign in to the Disaster Recovery Orchestrator Console UI using a web browser.

About the Firedrill operation

After configuring an application for DR, you might want to test whether the configuration works, without disrupting your production environment. To test a DR configuration, run a fire drill on it using the Disaster Recovery Orchestrator Console UI.

The following figure depicts the Firedrill view.



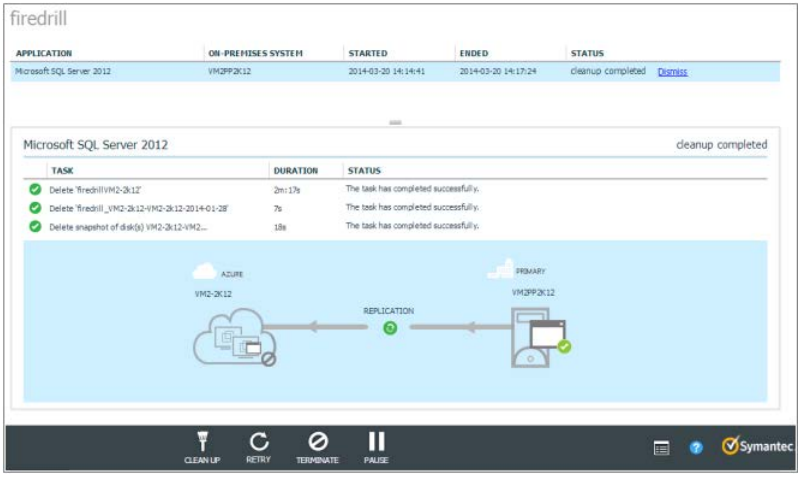
A fire drill comprises of the following tasks:

- Make a copy of the system disk of the cloud application host.
  This disk is retained in the Azure storage account after the virtual machine is deprovisioned as part of the recovery site preparation tasks.
- Take snapshots of the application data disks that are attached to the Console host.
  The application data is available on these disks, because it is replicated from the on-premises site to the cloud site.
- Create new disks in the cloud using these snapshots.
- Provision a new virtual machine with these disks in the fire drill network.
  This virtual machine is used for testing the Takeover operation, and is referred to as the fire drill application host.
- Prepare the application configuration on the fire drill application host.
- Bring the application online on the fire drill application host.
  The successful execution of this task indicates that the application can be available in the cloud environment if a failure occurs or after migration.
  During a fire drill, the status of the application at the on-premises site is not affected.

Note: All the resources that are used for a fire drill have the same names as their production counterparts, only prefixed with the text "firedrill". Thus, you can differentiate these resources from the resources that are used in your production environment in the Azure Management Portal.

After performing a fire drill, certain cleanup tasks need to be performed. Until you clean up the fire drill, the operation is considered incomplete.

The following figure depicts the tasks related to the Cleanup operation.



The fire drill cleanup tasks are as follows:

- Delete the virtual machine that was provisioned in the fire drill network.
- Delete the disks that were created for the fire drill configuration.
- Delete the snapshots that were created to test the DR configuration.

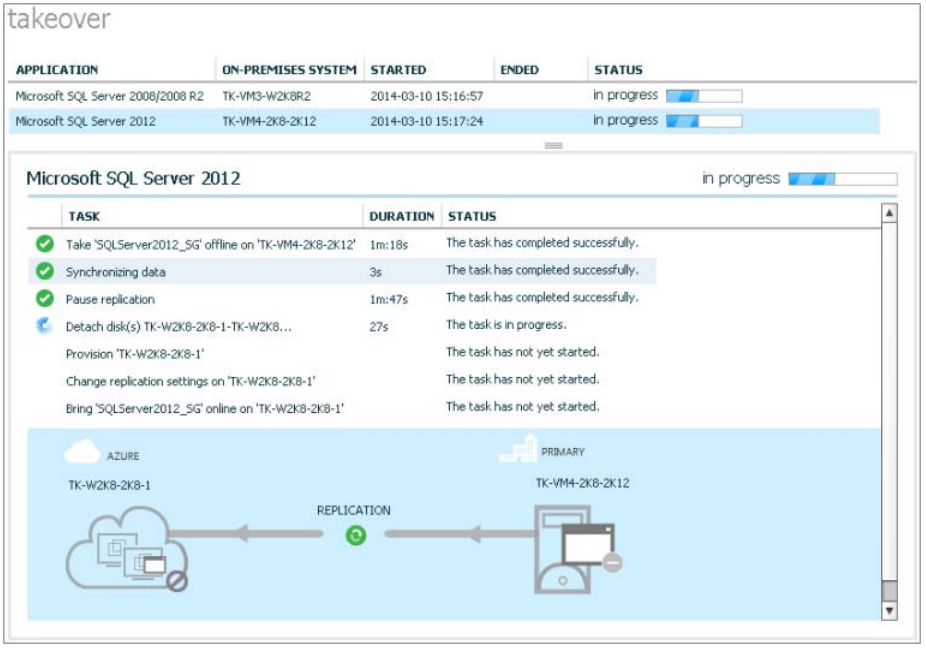
About the Takeover operation

Taking over application processing at the cloud site could be necessary in one of the following conditions:

- A disaster occurs at the on-premises site, causing the application to become unavailable.
- An infrastructure failure occurs at the on-premises site. For example, a fault occurs with the storage disks or the NICs.
- A routine maintenance activity is planned, during which, the on-premises application hosts need to be restarted.

- The application needs to be migrated to the cloud from the on-premises environment.

The following figure depicts the Takeover view of the Disaster Recovery Orchestrator Console UI.



A takeover operation comprises of the following tasks:

- Take the application offline on the on-premises application host.
- Complete replicating the application data from the on-premises storage to the cloud storage.
- Pause the replication.
- Detache the storage from the Console host.
- Provision the cloud application host.
- Change the replication settings on the cloud application host.
- Bring the application online on the cloud application host.

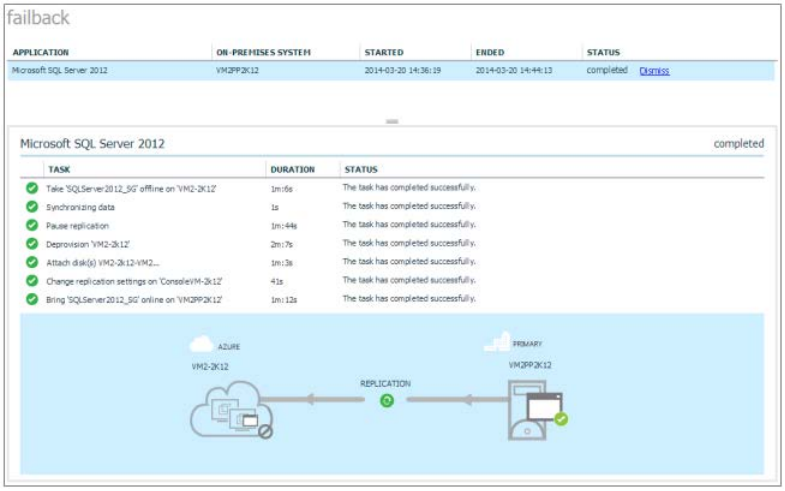
Note: While the takeover operation is in progress, the replication status appears as Unknown. The accurate replication status is displayed only after the operation is completed successfully.

About the Failback operation

Failing back application processing at the cloud site could be necessary in one of the following conditions:

- The on-premises site becomes available upon recovering from a disaster.
- An infrastructure failure at the on-premises site has been resolved, and the application processing can now be resumed. For example, a fault that occurred with the storage disks or the NICs at the on-premises site was resolved.
- A planned maintenance activity at the on-premises site has completed, and the system is available.
- The application needs to be migrated to the cloud from the on-premises environment.

The following figure depicts the Failback view of the Console UI.



A failback operation comprises of the following tasks:

- Take the application offline on the cloud application host.
- Synchronize the application data that was generated so far.
- Pause the replication.
- Deprovision the cloud application host.
- Attache the storage to the Console host.
- Change the replication settings on the Console host.
- Bring the application online on the on-premises application host.

Note: While the failback operation is in progress, the replication status appears as Unknown. The accurate replication status is displayed only after the operation is completed successfully.

About the replication of application data

Disaster Recovery Orchestrator conducts the file replication activities as follows:

- When you finalize an application recovery configuration, the replication of application data from the on-premises site to the cloud begins automatically.
- When you perform a takeover operation on the application, the direction of replication is reversed. While the takeover is in progress, the replication activity is paused. When the takeover is successful, the replication is resumed from the cloud to the on-premises site.
- When you perform a failback operation on the application, the direction of replication is reversed again. While the failback is in progress, the replication activity is paused. When the failback is successful, the replication is resumed from the on-premises site to the cloud.
- If the primary system restarts while the replication is in progress, the replication activity is stopped, regardless of the availability of the secondary system. When the primary system is online again, you need to start the replication from the Disaster Recovery Orchestrator Console UI.

  If you want the replication to start automatically after the primary system restarts, you must configure the **AutoStartReplication** tunable parameter.

  Note: Each time the replication starts, initial synchronization is done. This activity may take a long time if a large amount of data needs to be copied or if the network is slow.

- If the application is in the Stopped state when the primary system restarts, the replication is temporarily paused. When the primary system is online again, the replication is resumed. If you perform a planned restart of the primary system (on-premises application host or cloud application host), stop the application using the Health View. Doing so helps continue the replication quickly, because the initial synchronization is not required.
- If you encounter issues with the replication that are not resolved automatically within a reasonable period of time, you might want to stop the replication. After resolving such issues, you must start the replication again, so that the application remains recoverable in case of a site failure.

- If the replication log volume is unresponsive or detached for few moments, any application data that is written during that period is lost. To synchronize the application data correctly across both the sites, stop the replication and start it again.

For information about file replication:

See "File replication in Disaster Recovery Orchestrator" on page 94.

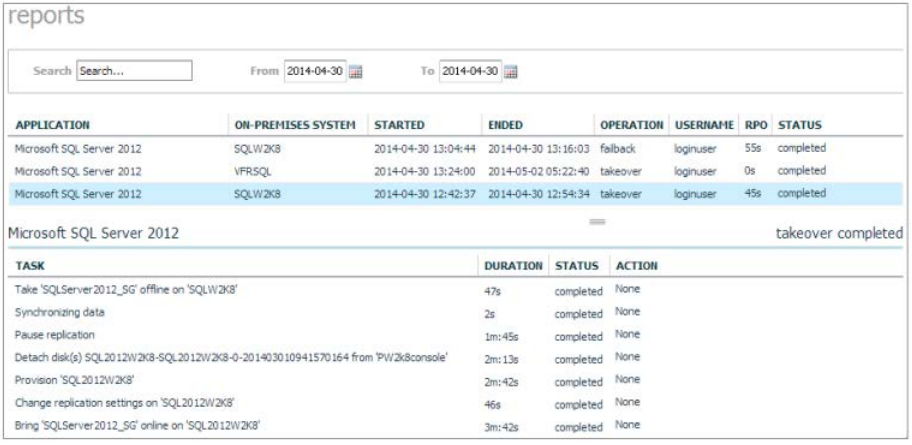
For information about stopping or starting the replication:

See "Starting or stopping the replication of application data" on page 52.

About Disaster Recovery Orchestrator reports

The Reports view of the Disaster Recovery Orchestrator Console UI lists all the operations performed on the applications configured for recovery in the cloud.

The following figure depicts the Reports view of the Console UI.



The Reports view provides the following information:

- Application name
- Name of the system on which the application was online when the operation was triggered
- Operation start time
- Operation end time
- Operation name
- Name of the user who performed the operation

- Recovery Point Objective (RPO)
  This is point in time upto which the application data can be recovered.
- Status of the operation

When you clicking on an application in the list, further details are displayed as follows:

- Tasks performed on the application configuration as part of the operation
- Duration in which the tasks were completed
- Status of the tasks
- Action

Additionally, you can perform the following operations in the Reports view:

- Print all the items displayed in this view or you filter the report and only print the required items
- Sort the table on one of the columns
- Filter the list based on the application name or the start and end time of the operation

About removing recovery configurations

To stop using the recovery services for an application, you must remove its recovery configuration. To remove a configuration, perform the unconfigure operation from the Disaster Recovery Orchestrator Console UI.

The unconfigure operation comprises of the following tasks:

- Take the application offline on the on-premises application host.
- Detach the storage from the Console host.
- Provision the cloud application host.
- Remove the application monitoring configuration from the on-premises application host and the cloud application host.
- Remove the file replication privileges.

See “Removing the recovery configuration of an application” on page 57.

Viewing a summary of all the applications recovery configurations

To view a summary of all the application recovery configurations, sign in to the Disaster Recovery Orchestrator Console UI as a guest user or a recovery administrator. The Dashboard view appears by default.

If you have a different view open, click the **Dashboard** menu in the navigation pane on the left.

The following figure depicts the Dashboard view.

The Dashboard view provides an overview of all the application recovery configurations on which you have guest or recovery administrator privileges. This view displays the following information:

- Total number of on-premises applications configured for recovery in the cloud
- Overview of applications at the on-premises site:
  - Number of applications in the following states: Online, Offline, Faulted, and Unknown
  - Number of Failback operations that are in progress or have failed
    When a Failback operation completes successfully, the application comes online at the on-premises site. Therefore, the number of successful Failback operations are not explicitly displayed.

- Number of applications for which the data replication is in the following states: Consistent, Inconsistent, Paused, or Stopped
- Overview of applications at the cloud site:
  - Number of applications in the following states: Online, Offline, Faulted, and Unknown
  - Number of Firedrill and Takeover operations that are in progress or have failed
    When a Firedrill operation completes successfully, the application comes online within the fire drill network at the cloud site. When a Takeover operation completes successfully, the application comes online at the cloud site. Therefore, the number of successful Firedrill operations or Takeover operations are not explicitly displayed.

Application states

The following table describes what the various application states mean.

Application State	Icon	Meaning
Online		The application is available and actively processing requests from the current site.
Offline		The application is not available at the current site.
Faulted		The application is available at the current site, but some of its components have faulted and so it is unresponsive. Check the application, its components, or the system that hosts the application to identify and resolve the issue.
Unknown		The application or its host is unresponsive. Check the application or its host to identify and resolve the issue.

Replication states

The following table describes what the various replication states mean.

Replication State	Icon	Meaning
Consistent		The initial synchronization of the application data between the on-premises site and the cloud site is complete. The data replication is in progress and is consistent.

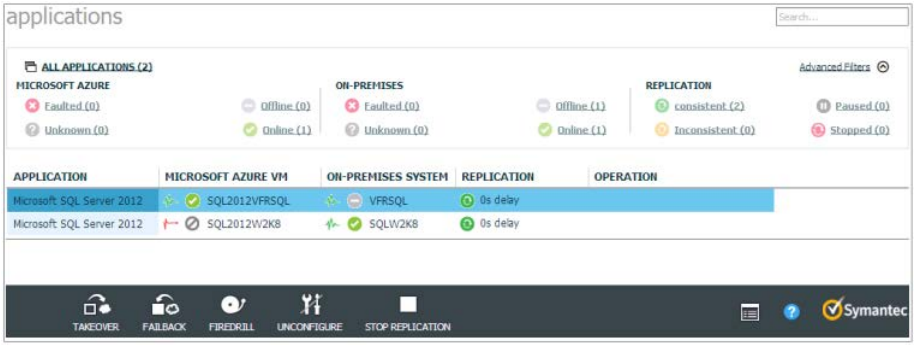
Replication State	Icon	Meaning
Inconsistent		The application is not available at the current site.
Paused		The replication is paused. Check whether the application, its components, or the system that hosts the application to identify and resolve the issue.
Stopped		The application or its host is unresponsive. Check the application or its host to identify and resolve the issue.

About working with application recovery configurations

To work with application recovery configurations, sign in to the Disaster Recovery Orchestrator Console UI as a guest user or a recovery administrator. Then, click the **Applications** menu in the navigation pane on the left. The adjacent number indicates the total number of applications that the signed-in user can view or administer.

The Applications view lists all the applications on which you have the guest or the recovery administrator privileges.

The following figure depicts the Applications view.



- If you sign in as a guest user, you can only view the available data, but you cannot perform any operations.
  See "Viewing the applications configured for DR" on page 44.
- If you sign in as a recovery administrator, you can configure or unconfigure applications for DR, and perform administrative operations on the applications.

See "Performing recovery operations and viewing the related information" on page 45.

The following information is displayed in the columns on the Applications view:

Column	Description
Application	Name of the application that is configured for DR
Microsoft Azure VM	■ Status of the application at the cloud site ■ Name of the cloud application host
On-Premises System	■ Status of the application at the on-premises site ■ Name of the on-premises application host
Replication	■ Status of data replication ■ Delay in replication
Operation	■ Name of an operation, if it has been triggered ■ Status of the ongoing operation ■ An action that you need to perform, if the operation has encountered an issue

The Applications view depicts a few more application states in addition to those depicted on the Dashboard view. The following table describes these states, which appear in the Microsoft Azure VM and the On-Premises System columns:

Application State	Icon	Meaning
Heartbeat: In Sync		The application is being monitored and appears healthy.
Heartbeat: Out of Sync		The application monitoring configuration has encountered an issue.
Unavailable		The application is unavailable, because the virtual machine is unavailable. This expected behavior occurs only in the case of the cloud application host, which is deprovisioned after the application recovery configuration is created. The virtual machine is provisioned again during the Takeover operation.
Configuration Incomplete		The application is configured for DR, but the application recovery configuration is not yet finalized.
Partially Available		The application is online, but some of its components are unresponsive.

This view also depicts a few more replication states in addition to those depicted on the Dashboard view. The following table describes these states, which appear in the Replication column:

Replication State	Icon	Meaning
Unkown		The replication file group (RFG) is configured, but replication is not yet started, or the replication service fails to identify the current state of replication.
Error		The replication service fails to receive a response from the secondary site, which possibly encountered an issue.

For information about the other application states and replication states:

See "Viewing a summary of all the applications recovery configurations" on page 40.

Viewing the applications configured for DR

When you open the Applications view, all the applications for which you have the guest user or the recovery administrator privileges are listed by default. You can filter the visible applications using the **Search** field or the **Advanced Filters** link.

For information about the operations that you can perform on the applications, refer to the following topic:

See "Performing recovery operations and viewing the related information" on page 45.

To view a subset of the applications based on predefined criteria

- Click the **Advanced Filters** link at the top-right corner of the Application view to display the predefined filters.

  Filter the applications as follows:

  - To view the applications that are in a specific state at the cloud site, click one of the links in the Microsoft Azure section. For example, click **Faulted** to view the applications that are online at the cloud site, but have encountered an issue.
  - To view the applications that are in a specific state at the on-premises site, click one of the links in the On-Premises section. For example, click **Unknown** to view the applications that were online at the on-premises site, but are currently unresponsive.
  - To view the applications for which the data replication is in a specific state, click one of the links in the Replication section. For example, click

Inconsistent to view the applications for which the data replication is in progress, but the data at both is not yet consistent.

The number adjacent to each link indicates the number of applications or the number replication activities in that corresponding state.

For information about the various application states and replication states, refer to the following topics:

See "Viewing a summary of all the applications recovery configurations" on page 40.

See "About working with application recovery configurations" on page 42.

To search within the visible applications list

- Search through the list of applications by entering one of the following values:
  - Application name
  - Cloud application host name
  - On-premises application host name
  - Operation

  If the value that you entered is part of any of these strings, the corresponding row is displayed.

  If a predefined filter is applied before entering the search value, the search is performed only on the filtered list of applications.

To view the complete list of applications

- To remove all the filters and view the complete list of configured applications, click the **All Applications** link at the top left corner of the Applications view.

  This link is useful in the scenarios where the applications list might appear empty, even though the application recovery configurations exist. For example, when you click an **in progress** link on the Dashboard view for the fire drill, takeover, or failback operations. Doing so opens the Applications view and displays only those applications on which the operation is in progress. After those operations are successful, the applications are removed from the list and it appears empty. Then, you can click **All Applications** to see the complete list.

Performing recovery operations and viewing the related information

A recovery administrator can perform various operations on the applications as described in the following procedure.

To perform an operation on an application

1 Select the application on which you want to peform an operation. If you click the application name, the application details view appears.

You can also select multiple applications and perform an operation on them simultaneously.

2 Click the appropriate button on the command bar as follows:

- **Firedrill**
  Click this button to test the application recovery configuration. The fire drill operation recovers the application in a bubble network in the cloud, which verifies that the recovery configuration works as expected.
  You can perform a fire drill only on an application that is currently online at the on-premises site.
  See "Testing a disaster recovery configuration by performing a fire drill" on page 48.
- **Start Replication** or **Stop Replication**
  When an application recovery configuration is finalized, the replication is started automatically. Click **Stop Replication** if you want to stop the ongoing replication to resolve an issue or to perform a planned activity that might disrupt the replication. To start the replication after it has been stopped (not paused), click **Start Replication**. The initial synchronization task is performed, which might take a long time to complete, depending on the data that needs to be copied from the primary site to the secondary site.
  See "Starting or stopping the replication of application data" on page 52.
- **Takeover**
  Click this button to take over the application processing in the cloud. This might be required in case of a failure at the on-premises site or in case of planned migration of the application to the cloud site.
  You can perform Takeover only on an application that is currently online at the on-premises site.
  See "Taking over application processing at the cloud site" on page 49.
- **Failback**
  Click this button to fail back the application processing to the on-premises site. You might want to do this when the on-premises has recovered from a failure, so that you do not incur costs for the resources in the cloud.
  You can perform Failback only on an application that is currently online in the cloud.
  See "Failing back application processing to the on-premises site" on page 51.
- **Unconfigure**

Click this button to remove the application recovery configuration.
You can remove an application recovery configuration only when the application is online at the on-premises site.
See "Removing the recovery configuration of an application" on page 57.

Note: If an operation is already in progress on some or all of the selected applications, an error message is displayed accordingly. You must wait until an ongoing operation is complete to trigger another operation.

Any signed-in user can do the following:

- View the complete list of applications, or apply the predefined filters, or search through the applications.
  See "Viewing the applications configured for DR" on page 44.
- View the logs that contain technical information about all the ongoing activities by clicking the Logs button at the bottom right corner.
- View the Disaster Recovery Orchestrator online help by clicking on the Help button at the bottom right corner.

Clearing the Faulted state of an application

If an application component faults, the application status appears as Faulted on the Applications view of the Console UI.

An application component may fault due to various reasons. Identify the issue and resolve it. Then, clear the Faulted state as described in the following procedure.

To clear the state of a Faulted application or its components

1. Log on to the system where the application appears Faulted.
2. Run the following command to identify the application configuration:

   ```
   hagrp -state
   ```

   Disaster Recovery Orchestrator displays the state of all the application monitoring components on the system, for example:

   ```
   #Group                          Attribute    System      Value
   InfrastructureSG_VM1P2K8        State        VM1P2K8     |ONLINE|
   SQLServer2008_SG                State        VM1P2K8     |OFFLINE|FAULTED|
   SQLServer2008_SG_805DBA2_SG     State        VM1P2K8     |ONLINE|
   ```

3 Run the following command on the application configuration:

```
hagrp -clear AppConfig
```

Replace the *AppConfig* variable with the name of the faulted application configuration, for example:

```
hagrp -clear SQLServer2008_SG
```

4 If the application state does not appear as Online on the Console UI after a few moments, launch the Health View, and click **Start Application**.

Testing a disaster recovery configuration by performing a fire drill

Run a fire drill on the recovery configuration to test whether the application can be recovered in the cloud. Running a fire drill does not disrupt your production environment, because the application is recovered in a separate bubble network.

To perform a fire drill

1 On the Applications view, select the application for which you want to test the DR configuration.

You can perform a fire drill on an application only after its DR configuration is complete and when it is available at the on-premises site.

2 On the command bar, click **Firedrill**.

3 When prompted to confirm whether to perform a firedrill operation on the application, click **Yes**.

On the Firedrill view, check the status of the tasks being performed for the selected application.

4 After all the the tasks are completed, click **Cleanup** on the command bar.

Unless you clean up the changes made during the fire drill, the operation is not considered to be complete.

5 When prompted to confirm whether you want to clean up the configuration, click **Yes**.

The Firedrill view indicates that the cleanup tasks is in progress.

6 After the cleanup tasks are completed, click the **dismiss** link for the application from one of the following locations:

- Status column of the Firedrill view
- Operation column of the Applications view

Enabling clients to access applications within a fire drill network

After performing the Firedrill operation on an application, the application comes online in the fire drill network. However, clients are unable to access the application, because its virtual name is not resolved to the cloud application host. To enable resolving application virtual names to the appropriate IP address in the fire drill network, perform the following steps.

To enable a client to access an application in a fire drill network

1. Log on to the cloud application host as a local administrator, and navigate to the following location:

   `C:\windows\System32\drivers\etc`

2. Open the `hosts` file using a text editor.
3. On a new line, add the IP address of the cloud application host, followed by at least one space, and then the application virtual name.

   For example:

   ```
   127.0.0.1 My_HR_Database
   ```

4. Save and close the `hosts` file.
5. Now, attempt to access the application from a client.

Taking over application processing at the cloud site

Take over application processing at the cloud site if the application becomes unavailable at the on-premises site or during a planned maintenance or migration.

To perform a takeover operation

1. On the Applications view, select the application that you want to begin processing from the cloud site.

   You can perform a takeover operation on an application only after the its DR configuration is complete and when it is unavailable at the cloud site.

2. On the command bar, click **Takeover**.
3. Click **Yes** when prompted to confirm whether to perform a takeover operation on the application.
4. On the Takeover view, check the status of the tasks being performed.

   Until the cloud application host is provisioned, the replication status appears as Unknown.

5. When the tasks are completed, click the **dismiss** link next to the application.

Updating on-premises application host when it comes online after takeover

When a failure occurs at the on-premises site, the application becomes unavailable. You perform the takeover operation so that the application processing can continue from the cloud site. The data replication state is changed to Stopped when the takeover operation completes, because the on-premises site is unavailable.

Later, when the on-premises site becomes available again, you need to prepare it for the failback operation. To do so, you update the premises configuration from the Console UI.

This operation comprises of the following tasks:

- Take the application offline at the on-premises site so that it does not conflict the corresponding instance that is currently online at the cloud site.
- Start the data replication from the cloud site to the on-premises site, which includes the initial synchronization task.
  This task may take a long time, depending on the amount of data that needs to be copied for the first time.

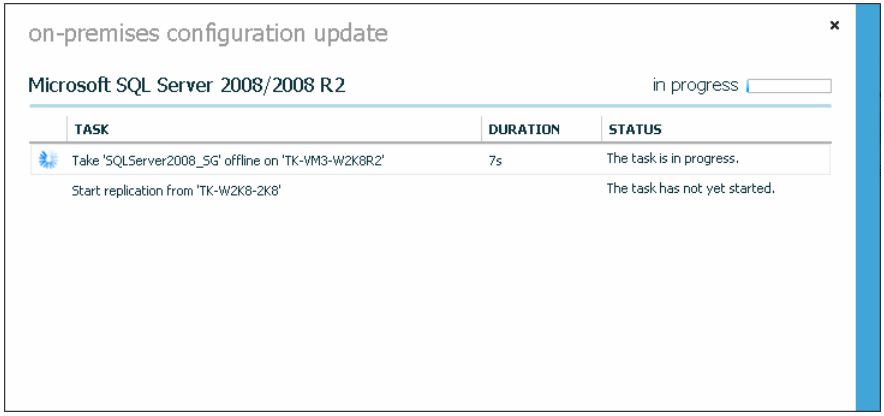
After you update the premises configuration, you can perform the failback operation successfully.

To update the on-premises application host

1 On the Applications view, select the application and click the **Update on-premises configuration** link in the Operation column.

2 When prompted to confirm whether you want to update the configuration, click **Yes**.

The On-Premises Configuration Update pop-up window displays the status of the tasks being performed.



3 When the tasks are completed, click the **dismiss** link in the Operation column.

The operation is then listed on the Reports view.

Failing back application processing to the on-premises site

After performing a takeover, the application becomes available at the cloud site. When the on-premises application host becomes available again, you need to restore the application to its original location.

To perform a failback operation

1. On the Applications view, select the application that you want to resume processing from the on-premises site.

   You can perform a failback operation on an application only when it is available at the cloud site and after the on-premises application host becomes available.
2. On the command bar, click **Failback**.
3. Click **Yes** when prompted to confirm whether to perform a failback operation on the application.
4. On the Failback view, check the status of the tasks being performed.

   Until the application becomes available again on the on-premises application host, the replication status appears as Unknown.
5. When the tasks are completed, click the **dismiss** link next to the application.

Starting or stopping the replication of application data

The replication of application data from the on-premises site to the cloud begins automatically when you finalize an application recovery configuration.

You might want to stop the replication if you encounter any replication issues that are not resolved automatically within a reasonable period of time. After resolving such issues, you must start the replication again, so that the application remains recoverable in case of a site failure.

Stopping or starting the replication is not allowed in the following conditions:

- The application recovery configuration has not been finalized.
- Any operation other than a fire drill is in progress.

Caution: When you stop an ongoing replication activity, the current state of the replication is lost. When you start the replication again, the initial synchronization task is performed. This task may take a long time, depending on the amount of data that needs to be copied for the first time.

Best practice

If you decide to stop an ongoing replication activity, create a snapshot of the target disks before you start the replication again. Doing so enables you to recover the application if a failure occurs while the initial synchronization task is in progress. If

no such snapshot exists and if a failure occurs during initial synchronization, you cannot recover the application at the secondary site.

To stop or start the replication

1. From the Applications view, select the application for which you want to stop or start the replication activity.
2. To stop the replication, click **Stop Replication** on the command bar.

   To start the replication again, click **Start Replication** on the command bar.

   This operation cannot be performed simultaneously on multiple applications. If you selected multiple applications, an error message is displayed. Dismiss the message, select a single application, and click the button again.

Warning: After you stop or start the replication from the Disaster Recovery Orchestrator Console UI, wait for the changed replication status to be reflected on the Applications view. If you want to perform any other operation on the application, do so only after the replication status is updated. Otherwise, you might encounter issues with the operation.

By default, if the primary system restarts, the replication is stopped and has to be started manually. You might want to configure Disaster Recovery Orchestrator to start the replication automatically after the primary system restarts. To do so, perform the following tasks on each on-premises application host and its corresponding cloud application host.

To enable Disaster Recovery Orchestrator to start the replication automatically after the primary system restarts

1. Open the Windows registry.
2. Navigate to the following key:

   `HKEY_LOCAL_MACHINE\SOFTWARE\SYMANTEC\VFR`
3. Create the DWORD value **AutoStartReplication**, and set its value data to **1**.
4. Save and close the registry.

For information about file replication and the related activities, refer to the following topics:

- See “File replication in Disaster Recovery Orchestrator” on page 94.
- See “About the replication of application data” on page 37.

For information about configuring an application for recovery in the cloud, refer to the following topics:

- See “Configuring an application for disaster recovery” on page 27.

- See "Finalizing the application recovery configuration" on page 30.

Changing the disaster recovery settings of an application

After you configure an application for recovery in the cloud, you can change some of its cloud site settings and replication settings.

To change the disaster recovery settings of an application

1. On the Applications page, click a cell in the Application column or double-click the application for which you want to change the settings.
2. Click the Settings tab to open it.
3. Make changes to the following fields as required:
   - Cloud site settings
     - Virtual Firedrill Network
       This drop-down list contains all the networks that are associated with your Azure subscription. You can select a different network to be used for testing the application recovery configuration.
     - Virtual Firedrill Subnet
       This drop-down list contains all the subnets that are associated with the selected network. You can select a different subnet to be used for testing the application recovery configuration.
     - Virtual Machine Size
       This drop-down list contains the various virtual machine sizes that are available in Azure. Select a different size to increase or decrease the processing and storage capability of the cloud application host as per your requirement.
       Changes made to the virtual machine size are reflected only when the next Takeover is performed.

       Note: If you resize a virtual machine from the Azure Management Portal, it restarts automatically. However, the changed virtual machine size does on reflect on the Console host.

       Resizing a virtual machine from the Azure Management Portal may impact the synchronization of data between the on-premises site and the cloud site. Therefore, Symantec recommends that you resize a virtual machine only from the Console UI.

- Replication settings
  - On-Premise Journal File Size
  - Azure Journal File Size

  Consider the following when changing the journal file sizes:
  - The size of the journal file must be at least 1 GB.
  - The sizes of both the journal files must match exactly.
  - When you save these changes, Disaster Recovery Orchestrator stops any ongoing replication activity, and the current state of the replication is lost. The replication is started again with the changed journal file sizes, and so the initial synchronization task is performed. This task may take a long time, depending on the amount of data that needs to be copied for the first time.

  Changes to the replication settings are not allowed in the following conditions:
  - The application recovery configuration has not been finalized.
    See "Configuring an application for disaster recovery" on page 27.
    See "Finalizing the application recovery configuration" on page 30.
  - Any operation other than a fire drill is being performed on the application.

4. On the action bar at the bottom of the page, click **Confirm** to save your changes.
5. On the confirmation message box that appears, click **OK**.

   If you do not want to save the chages made to this page, click **Cancel**, and then navigate to a different view.

Viewing reports of disaster recovery operations

Each time you perform a takeover, failback, or fire drill operation on an application, a report is generated automatically. This report lists the tasks that are performed as part of the operation.

Any user who is defined on the Privilege Settings view of the Disaster Recovery Orchestrator Console UI can view the reports of the disaster recovery (DR) operations performed on an application.

To view the disaster recovery reports of an application

1. Sign in to the Disaster Recovery Orchestrator Console UI using a web browser.
2. Open the Reports page to view a list of all the reports that are currently available. Each report corresponds to an operation that was performed on an application.
   - To view the report of a specific operation that was performed on an application, click the appropriate row in the table. The details of the tasks that were performed as part of that operation appear on the lower section of the pane. The tasks section provides the following information:
     - The description of the task
     - The duration in which the task was completed
     - The final status of the task
     - The action that was taken on the task, for example, if the task failed and was manually marked complete
   - To view the reports of operations based on specific criteria, one of the following values in the **Search** field:
     - Application name
       Displays the operations that were performed on this application
     - On-premises application host name
       Displays the operations that were performed on the application hosted on this system
     - Operation
       Displays the operations of this type that were performed on the applications
     - User name
       Displays the operations that were performed by this user
     - Status
       Displays the operations that were completed with this status

     The relevant reports are displayed in the table and the others are filtered out. To view a report, click the appropriate row in the table.
   - To view the reports that were generated between a certain number of days, specify the appropriate dates in the **From** and **To** fields.

Removing the recovery configuration of an application

When you want to stop using the recovery services for an application, you must remove the corresponding configuration.

Only recovery administrators can remove application recovery configurations. These users are configured with the Admin privileges on the Privilege Settings view of the Console UI.

To remove the recovery configuration of an application

1. Sign in to the Disaster Recovery Orchestrator Console UI using a web browser.
2. On the Applications view, select the application.

   Note: You can remove the recovery configuration only when the application is online at the on-premises site.

3. On the command bar, click **Unconfigure**.
4. Click **Yes** when prompted to confirm whether to remove the application recovery configuration.

   If you want to perform other tasks on the Console UI, close the Unconfigure wizard. To view the status of the unconfiguration tasks, click the **unconfigure in progress** link in the Operation column. If the operation has completed, this link appears as **unconfigure completed**.
5. If you want to remove another application recovery configuration, repeat the previous steps in this procedure for that application.
6. When the operation is successful:
   - If the Unconfigure wizard is open, click the check mark icon at its bottom right corner to exit.
   - If the wizard is closed, click the **Dismiss** link in the Operation column.

   The application no longer appears in the list.

   Note: The unconfiguration task does not reflect on the Reports view.

Chapter 4

Configuring application monitoring

This chapter includes the following topics:

Considerations for configuring an application for monitoring

Disaster Recovery Orchestrator provides an interface, Health View, to configure and administer application monitoring.

The Disaster Recovery Orchestrator Client installer creates a shortcut to the Health View on the system's desktop. The Health View is Web-based and can be accessed using any of the available browser.

You can also access the Health View directly from a browser window using the following URL:

`https://ClientHost:5634/vcs/admin/application_health.html`

Replace the *ClientHost* variable with the name of the system that hosts the application and Disaster Recovery Orchestrator Client. On the system itself, you may replace *ClientHost* with **localhost**.

Consider the following before you configure application monitoring:

- You can configure application monitoring on a system using the Symantec Application Monitoring Configuration Wizard. To launch the wizard, click **Configure Application Monitoring** on the Health View.

- You can use the wizard to configure monitoring for only one application on each system.
  To configure another application using the wizard, you must first unconfigure the existing application monitoring configuration.
- The wizard executes its tasks in the logged-on user context. Therefore, you must ensure that the logged-on user has administrative privileges on the system where you want to configure application monitoring.
- If you have configured a firewall, ensure that your firewall settings allow access to ports used by the Disaster Recovery Orchestrator installer, wizards, and services.
  For information about the ports used, refer to the *Symantec Disaster Recovery Orchestrator Deployment Guide*.
- After configuring services, processes, and mount points for monitoring, if you create another service, process, or mount point, then these new components are not monitored as part of the existing configuration.
  To monitor any new components that you add, unconfigure the existing application monitoring configuration and then run the wizard again to configure all the components.

  Note: When you configure or unconfigure application monitoring, it does not alter the state of the application. The application runs unaffected on the system.

- If you want to monitor storage that is managed using Symantec Storage Foundation (SFW), ensure that the volumes and mount points are created on dynamic disk groups.
  Disaster Recovery Orchestrator does not support monitoring for volumes and mount points created on cluster disk groups.
  This is applicable to on-premises systems only. On cloud virtual machines, the storage must be managed by Disaster Recovery Orchestrator only.

Configuring application monitoring

Perform the following steps to configure application monitoring on a system:

- Symantec recommends that you launch the Application Monitoring Configuration Wizard from within the Disaster Recovery Orchestrator Configuration Wizard. When the DR configuration wizard does not find an application monitoring configuration on the selected system, it displays a message box accordingly. Click **Configure** to launch the Symantec Application Monitoring Configuration Wizard.

- Alternatively, you may create the application monitoring configuration directly on the on-premises application host or the corresponding cloud application host. To do so, launch the Health View using the desktop shortcut or by entering the following URL in a browser:

  `https://system:5634/vcs/admin/application_health.html`

  Replace the *System* variable with the system name or its IP address. If you launch the browser locally on the system that hosts the application, you may replace *System* with **localhost**.

  Click **Configure Application Monitoring** to launch the Symantec Application Monitoring Configuration Wizard.

Note: You can configure monitoring for multiple services and processes in a single wizard workflow.

To configure application monitoring

1. Review the information on the Welcome panel and then click Next.
2. On the Application Selection panel, select the application that you want to configure for monitoring, and click Next.

   This panel lists all the applications on the system that are supported for monitoring. If the list of applications is too long, you might want to search for the application name using the **Search** box.
3. On the Windows Service Selection panel, select the services that you want to monitor.

   The wizard automatically discovers the services on the system.

   If a selected service depends on some other services, you must also select those services. You can define the dependencies between those services on the Start-Stop panel later.

   If you do not want to configure any services, click Next.
4. On the Windows Process Selection panel, specify the processes that you want to monitor.

   Perform the following steps sequentially to add a process:

   - Click **Add Process** to display the Process Parameters dialog box.
   - In the **Process Full Path** field type the complete path of the process executable file including its extension.

     If you define the process as a script (a Perl script, or a VBS script), specify the full path of the program that interprets the script (`perl.exe` or

`cscript.exe`) in the **Process Full Path** field and specify the full path of the script itself in the **Arguments** field.

For example, to specify `Perl.exe`, type the path as follows:

`C:\Program Files\Perl\Perl.exe`

- In the **Arguments** field, type the command line arguments for the process, if any.
- The specified process runs in the context of the local system account by default. To run the process in a different user's context, select the **Run process using specified credentials** check box and then specify the user name and password in the respective fields.

  The user name must be in the *user@domain.com* or*domain.com\username* format.
- Click **OK**.

  The process that you add is displayed on the wizard page.

  Repeat these steps for all the processes that you want to configure for monitoring.

If you do not want to configure any processes, click Next.

5 On the Mount Point Selection panel, select the mount points that you want to monitor.

If you do not want to monitor any mount points, click Next.

6 On the Define Start-Stop Order panel, specify the order in which you want the configured services, processes, and mount points to be started or stopped and then click **Configure**.

Perform the following steps sequentially to define the dependency between the components:

- Click on an application component name in the Parent Component box on the left.
- Select the check box for the desired component in the Component box on the right.

While starting the service or process, the components are brought online in the order that you specify here. For example, if a service is dependent on a mount point, then while starting the service the mount point is first brought online and then the service itself.

7 On the Application Monitoring Configuration panel, the wizard displays the tasks that are performed and the status of each task. After all the tasks are complete, click Next.

If the configuration tasks fail, click **View Logs** to check the details of the failure. Rectify the cause of the failure and run the wizard again to configure the application monitoring.

8 On the Finish panel, click **Finish** to exit the wizard.

This completes the application monitoring configuration.

Use the Health View to monitor the application status and control application monitoring.

Chapter 5

Administering application monitoring

This chapter includes the following topics:

About administering application monitoring

Disaster Recovery Orchestrator Client manages application monitoring, among other functions, on the on-premises application host and on the corresponding cloud application host. The on-premises application host can be a physical computer or a virtual machine. The cloud application host is a virtual machine.

Disaster Recovery Orchestrator provides an interface, Health View, to configure and administer application monitoring. The Disaster Recovery Orchestrator installation wizard creates a shortcut to the Health View on the system's desktop. The Health View is web based and can be accessed using any of the available browsers. You can also access the Health View directly from a browser window using the following URL:

`https://System:5634/vcs/admin/application_health.html?priv=ADMIN`

Replace the *System* variable with the name of the system that hosts the Client components.

Use the Health View to perform the following tasks:

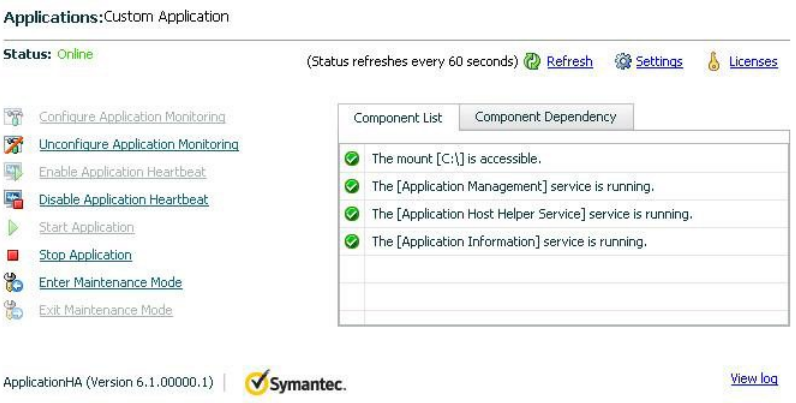
- Configure application monitoring
- Unconfigure application monitoring
- Start application
- Stop application
- Suspend application monitoring
- Resume application monitoring

You can also modify the application monitoring configuration settings using the Health View.

Viewing the status of configured applications

The Health View displays the status of configured applications, services or processes. It displays status in the following two views:

- **Component List**
  This view displays the list of services, processes, or mount points configured on the system for monitoring.
  The following figure represents a sample Health View, showing the component list for the configured services, processes, and mount points:

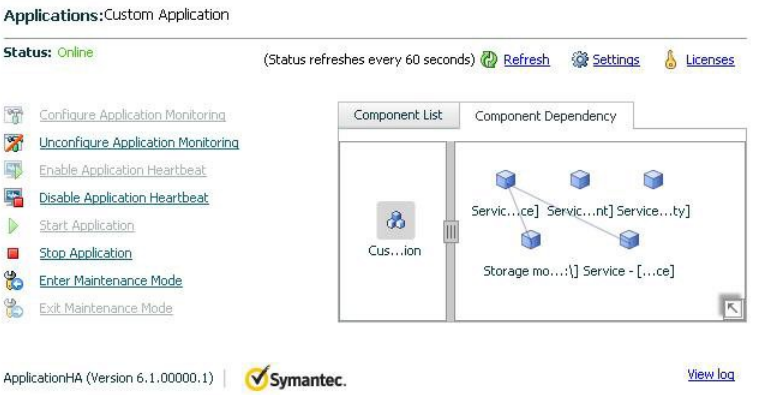


- **Component Dependency**

This view displays the dependency between the configured services, processes, and mount points.

The set dependency defines the order in which the components are started or stopped during a failure.

The following figure represents a sample Health View, showing the component dependency for the configured services, processes, and mount points:



The status is displayed as follows:

Online	Indicates that the services and processes are running on the system.
Offline	Indicates that the services and processes are not running on the system.
Partial	Indicates one of the following: ■ The services and processes are being started on the system. ■ Disaster Recovery Orchestrator was unable to start one or more of the configured services or processes.
Faulted	Indicates that the configured services or components have unexpectedly stopped running.

The status is refreshed every 60 seconds by default.

Starting or stopping the configured applications

Use the following options on the Health View to start or stop the configured application and the associated components:

- Click **Start Application** to start a configured application.

Disaster Recovery Orchestrator attempts to start the configured application and its components in the required order. The configured components are brought online in the appropriate hierarchy.

- Click **Stop Application** to stop a configured application that is running on the system.
  Disaster Recovery Orchestrator begins to stop the configured application and its components gracefully. The configured resources are taken offline in the predefined order.

Suspending or resuming application monitoring

After configuring application monitoring you may want to perform routine maintenance tasks on those applications. These tasks may or may not involve stopping the application but may temporarily affect the state of the applications and its dependent components. If there is any change to the application status, Disaster Recovery Orchestrator Client may try to restore the application state. This may potentially affect the maintenance tasks that you intend to perform on those applications.

If stopping the application is not an option, you can suspend application monitoring and create a window for performing such maintenance tasks. When application monitoring is suspended, the Client freezes the application configuration.

- Click **Enter Maintenance Mode** to suspend the application monitoring for the applications that are configured on the system. During the time the monitoring is suspended, the Client does not monitor the state of the application and its dependent components.
  The Health View does not display the current status of the application. If there is any failure in the application or its components, the Client does not take any action.
- Click **Exit Maintenance Mode** to resume the application monitoring for the applications configured on the system. You may have to click the **Refresh** link to see the current status of the application.
  If you have made changes that include database addition or change in the underlying storage mount point that was being monitored, then those changes may not reflect in the application monitoring configuration. In such cases, you may have to unconfigure and reconfigure the application monitoring.

Administering application monitoring settings

The Health View provides a set of options that you can use to control the way Disaster Recovery Orchestrator Client handles the following functions on the system:

- Application monitoring
- Applications and dependent component faults
- Application recovery

These configuration settings are applicable on a per system basis. The settings apply to all the applications that the Client monitors on the system.

The following settings are available:

- **App.RestartAttempts**
  This option defines the number of times that the Client should try to restart a failed application or its dependent component.
  If an application fails to start in the specified number of attempts, the Client sets its status to Faulted.
- **App.ShutdownGraceTime**
  This option defines the number of seconds Disaster Recovery Orchestrator Client should wait before setting the application status Faulted.
  AppShutDownGraceTime value can vary between 0 and 600. The default is 300 seconds.
- **App.StartStopTimeout**
  When you click the **Start Application** or **Stop Application** links on the Health View, the Client initiates an orderly start or stop of the application and its dependent components. This option defines the number of seconds the Client must wait for the application to start or stop. If the application does not respond in the stipulated time, an error is displayed on the Health View.
  A delay in the application response does not indicate that the application or its dependent component has faulted. Parameters such as workload, system performance, and network bandwidth may affect the application response. The Client continues to wait for the application response even after the timeout interval is over. If the application fails to start or stop, the Client takes the necessary action depending on the other configuration settings.
  AppStartStopTimeout value can vary between 0 and 600. The default is 30 seconds.

To modify the application monitoring configuration settings

1. Launch the Health View using the following URL:

   `https://System:5634/vcs/admin/ application_health.html?priv=ADMIN`

   Replace the *System* variable with the name of the system that hosts the application and the Disaster Recovery Orchestrator Client components.
2. Click the **Settings** link to display the Settings dialog box.
3. Specify the values for the available options displayed in the Settings dialog box and then click **OK**.

   The specified values are updated in the configuration, and they take effect immediately.

Restarting an application host

You might occasionally need to restart the application host systems, for example, after applying updates for the operating system. To do so, perform the following steps.

To restart the on-premises application host or the cloud application host

1. Launch the Health View, click **Stop Application**, and close the Health View.

   Disaster Recovery Orchestrator Client takes the application monitoring configuration and the corresponding replicated file group (RFG) offline.
2. Restart the system.
3. Launch the Health View, and click **Start Application**.

   Disaster Recovery Orchestrator Client brings the application monitoring configuration and the corresponding replicated file group (RFG) online.

Unconfiguring application monitoring

Disaster Recovery Orchestrator removes an application monitoring configuration when you remove the application recovery configuration.

You might want to directly remove an application monitoring configuration to re-create the configuration in case of issues or to configure another application for monitoring. To remove the application monitoring configuration directly from a system, use the Health View.

Note: If you remove the monitoring configuration of an application that is also configured for recovery, the recovery configuration becomes invalid.

To unconfigure application monitoring

1. Log on to the system where the application is configured for monitoring, and access the Health View.
2. Click **Unconfigure Application Monitoring**.

   Disaster Recovery Orchestrator removes all the configured resources for the application and its services.

Note: Removing an application monitoring configuration does not uninstall Disaster Recovery Orchestrator Client from the system. The unconfigure option removes all the application monitoring configuration resources from the system. To monitor an application, you have to configure them again.

Appendix A

Symantec Disaster Recovery Orchestrator agents

This appendix includes the following topics:

AppStatusHB (heartbeat) agent

The heartbeat agent, AppStatusHB, monitors the health of the application configured for application monitoring and disaster recovery (DR). To configure an application for DR, you use the Disaster Recovery Orchestrator Configuration Wizard. The wizard creates a resource for the AppStatusHB agent, which monitors the status of the configured application and relays that information to Disaster Recovery Orchestrator Console.

The `AppStatusHB` resource type represents this agent.

AppStatusHB agent function

Monitor	Determines the status of the specified application configuration regardless of the site where the application is running and relays it to Disaster Recovery Orchestrator Console.

AppStatusHB agent state definitions

ONLINE	Indicates that the agent is monitoring the configured application correctly.
OFFLINE	Indicates that the agent is not monitoring the configured application.
UNKNOWN	Indicates that the agent cannot determine the status of the AppStatusHB resource. This might mean that the resource was configured incorrectly.

AppStatusHB agent resource type definition

```
type AppStatusHB (
  static i18nstr ArgList[] = { ConsoleIP, Site, SGProps }
  str ConsoleIP
  str Site
  str SGProps{}
)
```

Sample AppStatusHB resource

A sample configuration of the AppStatusHB resource is as follows:

```
AppStatusHB Application_SG_AppStatusHB (
  ConsoleIP = "10.217.169.90"
  Site = premise
  SGProps = { Application_SG = Application_SG_E13AB37 }
)
```

AppStatusHB agent attributes

Table A-1

Attribute	Description
Name: **ConsoleIP** Type: **String** Dimension: **Scalar**	The Disaster Recovery Orchestrator Console IP address.

Table A-1 *(continued)*

Attribute	Description
Name: **Site** Type: **String** Dimension: **Scalar**	If this attribute is set to `premise`, it indicates that resource is configured for the on-premises system. If this attribute is set to `cloud`, it indicates that resource is configured for a virtual machine in the cloud.
Name: **SGProps** Type: **String** Dimension: **Vector**	The application configuration that is being monitored, along with the unique resource ID generated by the Disaster Recovery Orchestrator Configuration Wizard.

MountMonitor agent

The MountMonitor agent monitors the mount path of the configured storage. It is independent of how the underlying storage is managed. The mount path can be a drive letter or a folder mount.

When configuring a directory to host the mount, verify the following conditions:

- The configured path exists.
- The directory is empty.
- The volume on which the directory resides is NTFS-formatted.

The `MountMonitor` resource type represents this agent.

MountMonitor agent function

Online	Mounts the configured mount path (drive letter or folder) on the system.
Monitor	Verifies that the specified mount path (drive letter or folder) is mounted.

MountMonitor agent state definitions

ONLINE	Indicates that the system can access the configured mount path.
OFFLINE	Indicates that the system cannot access the configured mount path.
UNKNOWN	Indicates that the agent cannot determine the status of the resource. This might mean that the application was configured incorrectly.

MountMonitor agent resource type definition

```
type MountMonitor (
  static int IMF{} = { Mode=3, MonitorFreq=5, RegisterRetryLimit=3 }
  static i18nstr IMFRegList[] = { MountPath, VolumeName }
  static i18nstr ArgList[] = { MountPath, VolumeName, MountDependsOn }
  static str Operations = OnOnly
  str MountPath
  str VolumeName
  str MountDependsOn{}
)
```

Sample MountMonitor resource

A sample configuration of the MountMonitor resource is as follows:

```
MountMonitor MountMonitor_F (
  MountPath = "F:\\"
  VolumeName = "\\\\?\\Volume{cee30074-f32f-11e2-941e-005056b46977}\\"
)
```

MountMonitor agent attributes

Table A-2 Required MountMonitor agent attributes

Attribute	Description
Name: **MountPath** Type: **String** Dimension: **Scalar**	The drive letter or path to an NTFS folder where a partition is mounted. The attribute can be specified in one of the following formats: ■ `X` ■ `X:` ■ `X:\` ■ `X:\Directory` ■ `X:\Directory\`
Name: **VolumeName** Type: **String** Dimension: **Scalar**	The GUID of the volume to be mounted.

Table A-2 Required MountMonitor agent attributes *(continued)*

Attribute	Description
Name: **MountDependsOn** Type: **String** Dimension: **Scalar**	Defines the dependency between the nested mount points. If the application data is stored on nested mount points, then it is required to set the dependency between these mount points. This enables Disaster Recovery Orchestrator Client to monitor all the nested mount points. If this attribute is not configured, Disaster Recovery Orchestrator only monitors the last mount point. The value of this attribute must be specified as a key-value pair, where: Key = *MountPath* Value = *VolumeName*

IPv4Monitor agent

The IPv4Monitor agent monitors the static or dynamic IP address assigned to the network interface card (NIC).

The IPv4Monitor agent supports intelligent resource monitoring and uses Intelligent Monitoring Framework (IMF) for resource state change notifications. The agent relies on the network and hardware events raised by the operating system. For example, the operating system raises an event when an IP address becomes unavailable.

The `IPv4Monitor` resource type represents the IPv4Monitor agent, and it is a persistent resource.

IPv4Monitor agent function

Monitor — Determines whether the NIC specified in the MACAddress attribute is static or dynamic:

- If the NIC is static, it monitors the IP address mentioned in the Address attribute.
- If the NIC is dynamic, it retrieves the IP address assigned to the NIC and monitors it.

IPv4Monitor agent state definitions

ONLINE — Indicates that the IP address assigned to the NIC is available.

UNKNOWN	Indicates that the agent encountered errors while monitoring the IP address. This might mean that the resource was configured incorrectly.
FAULTED	Indicates that IP address being monitored is unavailable (static) or has been changed (dynamic).

IPv4Monitor agent resource type definition

```
type IPv4Monitor (
  static int IMF{} = { Mode=3, MonitorFreq=5, RegisterRetryLimit=3 }
  static i18nstr IMFRegList[] = { Address, MACAddress }
  static i18nstr ArgList[] = { Address, SubNetMask, MACAddress, IsCloudVM }
  static str Operations = None
  str Address
  str SubNetMask
  str MACAddress
  boolean IsCloudVM = 0
)
```

Sample IPv4Monitor resource

```
IPv4Monitor Application_SG_IPv4Monitor (
  Address = "10.217.52.75"
  SubNetMask = "255.255.252.0"
  MACAddress = "00:50:56:B4:69:77"
  IsCloudVM = 1
)
```

IPv4Monitor agent attributes

Table A-3 Required IPv4Monitor agent attribute

Attribute	Description
Name: **MACAddress** Type: **String** Dimension: **Scalar**	The physical address of the NIC to which the IP address is assigned Use the `ipconfig -all` command to retrieve the physical address of a NIC.

Table A-4 Optional IPv4Monitor agent attributes

Attribute	Description
Name: **Address** Type: **String** Dimension: **Scalar**	A unique IP address assigned to the NIC. This attribute is optional for DHCP-enabled NICs.
Name: **SubNetMask** Type: **String** Dimension: **Scalar**	The subnet mask associated with the IP address. This attribute is optional for DHCP-enabled NICs.
Name: **IsCloudVM** Type: **Boolean** Dimension: **Scalar**	If this attribute is set to 1, it indicates that the resource is configured for a cloud virtual machine. If this attribute is set to 0 (zero-default), it indicates that the resource is configured for an on-premises system.

Lanman agent

The Lanman agent enables clients to access data and applications on a system by associating its IP address with the virtual computer name in the WINS database. The agent provides the option of associating multiple IP addresses from different subnets with the virtual computer name. The agent also provides the option of creating the virtual computer object in any organizational unit in the Active Directory and enhances the DNS updating capabilities of Disaster Recovery Orchestrator.

The Lanman agent registers the following services with the WINS server:

- Server (20h)
- Workstation (00h)
- Messenger (03h)

The agent supports Kerberos authentication by providing the option of adding the virtual computer name to the Active Directory and adding the system's IP address to the DNS. The agent uses the VCS Helper Service user context for AD and DNS updates.

The Lanman agent updates and monitors the canonical name (CNAME) mapping in the domain name server when failing over applications across subnets (performing a wide-area failover.) The Lanman agent also supports creating DNS records in different DNS zones.

DNS scavenging affects virtual servers configured in Disaster Recovery Orchestrator because the Lanman agent uses DNS to map virtual names with IP addresses. If

you use scavenging, then you must set the DNSRefreshInterval attribute. This will enable the Lanman agent to refresh the resource records on the DNS servers. See the DNSRefreshInterval attribute description for more information.

If security policies are enabled on the Windows Server, ensure that the startup type of the Server Service is set to Automatic.

Dependency of Lanman agent

The Lanman resource depends on the IPv4Monitor resource.

If you change your Lanman resource dependency to a new IPv4Monitor resource and bring the Lanman resource online, a ping to the virtual name might respond from the IP address of the previous IPv4Monitor resource until the next WINS broadcast. The WINS broadcast updates the WINS database with the changed association.

For example, if you took the Lanman resource offline, changed the Lanman resource dependency from IP_A to IP_B, and brought the Lanman resource online, a ping to the virtual name might still respond from IP_A. Note that the IP_A resource is kept online during this process. The ping will respond from IP_B after the next WINS broadcast updates the WINS database.

Lanman agent functions

Online	Binds the IP addresses with the specified virtual computer name. The agent also queries the name server of the domain for Host (A), PTR, and CNAME records and adds or updates the records on the name server.
Offline	Removes the IP address binding from the virtual computer name.
Monitor	Verifies the IP addresses are bound to the virtual computer name. If DNSUpdateRequired and DNSRefreshRequired is enabled and the resource is online, then the Lanman agent refreshes the resource records on the DNS servers. The agent queries the name servers for DNS records. It reports back ONLINE if the response from all the name servers contains the Host (A), PTR, and CNAME records. If no servers return the appropriate records, the monitor reports the resource as OFFLINE.

Lanman agent state definitions

ONLINE	Indicates that the IP addresses are bound to the virtual computer name and the DNS records are as expected.

OFFLINE	Indicates one or more the following: ■ The IP addresses are not bound to the virtual computer name. ■ The agent failed to create the DNS records. ■ The expected DNS records were not found.
UNKNOWN	Indicates that the agent could not determine the status of the resource. This might mean that the resource was configured incorrectly.

Lanman agent resource type definition

```
type Lanman (
  static i18nstr ArgList[] = { VirtualName, MultiNet, "IPResName:Address",
    "IPResName:SubNetMask", "IPResName:MACAddress", "IPResName:Prefix",
    MultiNetInfo, DNSUpdateRequired, ADUpdateRequired, DNSCriticalForOnline,
    ADCriticalForOnline, ADContainer, DNSOptions, AdditionalDNSServers,
    DNSRefreshInterval, DNSZones, AliasName, TSIGKeyFile, TTL }
  str VirtualName
  str IPResName
  boolean MultiNet = 0
  str MultiNetInfo[]
  boolean DNSUpdateRequired = 0
  boolean ADUpdateRequired = 0
  boolean DNSCriticalForOnline = 0
  boolean ADCriticalForOnline = 0
  str ADContainer
  str DNSOptions[]
  str AdditionalDNSServers{}
  int DNSRefreshInterval
  str DNSZones{}
  str AliasName
  str TSIGKeyFile
  int TTL
)
```

Sample Lanman resource

A sample configuration of the Lanman resource is as follows:

```
Lanman Application_SG_Lanman (
  VirtualName = TL01
  IPResName = Application_SG_IPv4Monitor
  DNSUpdateRequired = 1
```

```
    DNSCriticalForOnline = 1
)
```

Lanman agent attributes

Table A-5 Required Lanman agent attributes

Attribute	Description
Name: **IPResName** Type: **String** Dimension: **Scalar**	The name of the IPv4Monitor resource on which the Lanman resource depends. Do not define a value for this attribute if the MultiNet attribute is set to 1.
Name: **VirtualName** Type: **String** Dimension: **Scalar**	The virtual computer name to be assigned to the server. The virtual name must be fewer than 15 characters. Note that if you specify a virtual computer name in lowercase letters, the agent converts it to uppercase. For example, the name VCSServer is converted to VCSSERVER.

Table A-6 Optional Lanman agent attributes

Attribute	Description
Name: **ADContainer** Type: **String** Dimension: **Scalar**	Specifies the distinguished name of the Active Directory container or the organizational unit (OU) for the newly created computer object. If no value is specified for this attribute, the Lanman resource creates the computer object in the default container "Computers." Note that the user account for VCS Helper service must have adequate privileges on the specified container to create and update computer accounts. Refer to Microsoft documentation for information on assigning user privileges for a container. By default, the attribute contains no value. **Note:** Value specified for this attribute will be effective only if ADUpdateRequired is set to 1.
Name: **ADCriticalForOnline** Type: **Boolean** Dimension: **Scalar**	Defines whether the Lanman resource faults if the agent fails to update the Active Directory. The value 1 indicates that the resource faults in case of a failure to update the Active Directory. The value 0 indicates that it does not. Default: 0 (zero)

Table A-6 Optional Lanman agent attributes *(continued)*

Attribute	Description
Name: **AdditionalDNSServers** Type: **String** Dimension: **Association**	An array that specifies the IP addresses of the additional DNS servers that will be updated by the Lanman resource. For all the Windows DNS servers, the forward and reverse lookup zones must be configured. For all the Berkeley Internet Name Domain (BIND) servers, only the forward lookup zones are required. All additional DNS servers are considered as Windows DNS servers by default. If any additional DNS server is a Berkeley Internet Name Domain (BIND) server, you will have to specify it in the attribute value. Example: `"{"10.212.108.9" = "","10.212.108.10" = "BIND"}"` Where 10.212.108.9 is the IP address of a WindowsDNSserver, and 10.212.108.10 is the IP address of a BIND DNS server. By default, the attribute contains no value. Values specified for this attribute will be effective only if DNSUpdateRequired is set to 1. The Lanman agent creates only CNAME records on BIND servers. You must also specify the AliasName attribute in case of BIND server updates. **Note:** The Lanman agent supports BIND version 8 and above. **Note:** In cases where the default DNS is a BIND DNS server, set the value of the DNSOptions attribute to IgnoreDefault, and specify the BIND DNS server details in this attribute. **Note:** If the BIND DNS servers are configured for secure updates, then you must configure the TSIG keys either in the DNSZones attribute or the TSIGKeyFile attribute.
Name: **ADUpdateRequired** Type: **Boolean** Dimension: **Scalar**	Defines whether the Lanman resource updates the Active Directory with the virtual name. The value 1 indicates that the agent updates the Active Directory. The value 0 indicates it does not. Default is 0.
Name: **DNSCriticalForOnDimension** Type: **Boolean** Dimension: **Scalar**	Defines whether the Lanman resource faults if the agent fails to update the DNS. The value 1 indicates that the resource faults in case of a failure to update the DNS. The value 0 indicates that it does not. Default is 0.

Table A-6 Optional Lanman agent attributes *(continued)*

Attribute	Description
Name: **DNSOptions** Type: **String** Dimension: **Vector**	An array that specifies the way in which the Lanman resource updates the DNS servers. This attribute can take one or all of the following values: ■ UpdateAll: Updates all the default DNS servers specified in the TCP/IP properties for the system, and the additional DNS servers specified in the AdditionalDNSServers attribute. ■ IgnoreDefault: Ignores the default DNS servers and updates only the additional DNS servers. ■ PurgeDuplicate: Removes duplicate DNS entries from the DNS servers. Symantec recommends you set this value for applications configured for wide area failover. ■ SkipPtrRecords: The Lanman resource excludes the PTR records while updating the resource records on the specified DNS servers. Any combination of these values can be specified for the attribute. This attribute takes effect only when the Lanman resource comes online. See "Updating manual DNS entries" on page 84. By default, the attribute contains no value. Values specified for this attribute will be effective only if DNSUpdateRequired is set to 1 and additional DNS servers are specified in the AdditionalDNSServers attribute. **Note:** In cases where the default DNS is a BIND DNS server, set this attribute value to IgnoreDefault, and specify the BIND DNS server details in the AdditionalDNSServers attribute.
Name: **DNSUpdateRequired** Type: **Boolean** Dimension: **Scalar**	Defines whether the Lanman resource updates the DNS with the virtual IP address. The value 1 indicates that the resource updates the DNS. The value 0 indicates it does not. Default is 0. If you set this attribute but there are no DNS servers specified in the TCP/IP properties, then you must specify the DNS servers that you wish to update in the AdditionalDNSServers attribute. If NetBIOS is disabled over TCP, set this attribute value to 1. **Note:** The Lanman resource does not update the DNS for the manual DNS entries. See "Updating DNS servers" on page 84.

Table A-6 Optional Lanman agent attributes *(continued)*

Attribute	Description
Name: **DNSRefreshInterval** Type: **Integer** Dimension: **Scalar**	This attribute represents the time interval, in seconds, after which the Lanman agent attempts to refresh the resource records (RRs) on the DNS servers. You must set a value for this attribute if you want the Lanman agent to refresh the records on the DNS servers. The default value zero indicates that the Lanman agent is unable to refresh the DNS records, and the records are removed as a result of a scavenging operation or by the DNS administrator, the Lanman resource will fault.
Name: **DNSZones** Type: **String** Dimension: **Association**	An array that specifies a list of DNS zones (in case of multi-domain environments with parent-child configurations) for which the Lanman resource should create and update Address (A) records and canonical name (CNAME) records in the DNS server of the parent domain. Example: {"child1.company.com", "child2.company.com"} Where child1.company.com and child2.company.com are DNS zones representing different child domains. By default, the attribute contains no value. This means that the Lanman agent will create and update resource records only in the DNS name servers for the zones in which the nodes exist. If multiple zones are being updated on BIND DNS servers that are configured for secure updates, then each zone may require a different TSIG key. In such a case, you must specify the absolute path of the TSIG key file in the attribute value. Example: {"child1.company.com" = "C:\TSIGKey1.key", "child2.company.com" = "C:\TSIGKey2.key"} Where TSIGKey1.key is the TSIG key for the DNS zone child1.company.com, and TSIGKey2.key is the TSIG key for the DNS zone child2.company.com. **Note:** The Lanman agent supports BIND version 8 and above.

Table A-6 Optional Lanman agent attributes *(continued)*

Attribute	Description
Name: **AliasName** Type: **String** Dimension: **Scalar**	A string representing the alias to the canonical name. The Lanman agent creates a CNAME record using the value specified in this attribute. Example: "www" Where www is the alias to the canonicalnamemtv.veritas.com. By default, the attribute contains no value. **Note:** This attribute is required if a BIND DNS server is specified in the AdditionalDNSServers attribute.
Name: **TSIGKeyFile** Type: **String** Dimension: **Scalar**	Required when you configure BIND DNS for secure updates. Specify the absolute path to the file that contains the private Transaction Signature (TSIG) key. This key is used by the `nsupdate` utility to perform secure BIND DNS updates. See the BIND man pages for more information about secure DNS updates. You must copy the files containing the keys (typically the .key and the .private file) on each of the nodes that is listed in the system list for an application configuration. By default, the attribute contains no value. Example: C:\TSIG\Kveritas.com.+157+00000.key **Note:** The Lanman agent supports BIND version 8 and above.
Name: **TTL** Type: **Integer** Dimension: **Scalar**	This value defines the Time To Live (TTL) value (in seconds) that gets stored in the DNS records created by the agent. Default: 0 Example: TTL = 7200
Name: **MultiNet** Type: **Boolean** Dimension: **Scalar**	Defines whether the Lanman resource binds multiple IP addresses with the virtual name. The value 1 indicates the resource binds multiple IP addresses specified in MultiNetInfo with the virtual computer name. The value 0 indicates the resource binds a single IP address specified in IPResName. Default is 0.

Table A-6 Optional Lanman agent attributes *(continued)*

Attribute	Description

Name: **MultiNetInfo**
Type: **String**
Dimension: **Vector**
	An array that specifies details of the IP addresses to be bound to the virtual computer name. If MultiNet is set to 1, configure this attribute in the following format:
	✓
	`MultiNetInfo = { "IP=ip_address1`
	`Mask=subnetmask1`
	`WINS=wins_ip_address1`
	`MACAddress=macaddress1",`
	`"IP=ip_address2`
	`Mask=subnetmask2`
	`WINS=wins_ip_address2`
	`MACAddress=macaddress2" }`
	Note: Specifying Mask and MACAddress is optional. If not specified, the Lanman agent discovers the subnet mask from the current configuration.
	Note: MACAddress is required if netbios is disabled for the IP address, on Windows Server 2008 only.

Updating manual DNS entries

Perform the following steps to update the DNS for manual DNS entries.

To update the DNS for manual DNS entries

1. For the manually added DNS entry, add the user in whose context the VCS Helper service is running.
2. Assign "Full Control" privilege to the newly added user.

Refer to Microsoft documentation for information about adding users and assigning privileges.

Updating DNS servers

Table A-7 presents possible combinations of values for the DNSOptions attribute and the updates effected by the Lanman resource corresponding to each value set.

Table A-7 DNSOptions attribute and Lanman agent behavior

UpdateAll	Ignore Default	Purge Duplicate	Effect
-	-	-	Updates any one default DNS server.
-	-	✓	Updates any one default DNS server and removes duplicate entries, if any.
-	✓	-	Updates any one additional DNS server.
-	✓	✓	Updates any one additional DNS server and removes duplicate entries, if any.
✓	-	-	Updates all the default and additional DNS servers.
✓	-	✓	Updates all the default and additional DNS servers and removes duplicate entries, if any.
✓	✓	-	Updates all the additional DNS servers.
✓	✓	✓	Updates all additional DNS servers and removes duplicate entries, if any.

GenericService agent

The GenericService agent brings services online, takes them offline, and monitors their status. A service is an application type that is supported by Windows, and conforms to the interface rules of the Service Control Manager (SCM).

Services are configured as resources of type GenericService. You can configure the GenericService agent to monitor multiple services by defining a resource for each service to be monitored. You can monitor a service in a user-context by specifying the user name, password, and domain.

Note: The service to be configured using the GenericService agent must not be in a disabled state.

This agent is represented by the GenericService resource type.

This agent supports intelligent resource monitoring and uses Intelligent Monitoring Framework (IMF) for resource state change notifications. The agent traps the Windows service related events and takes appropriate action if a configured service stops or fails to respond.

Agent functions

Online	Starts the configured service.
Offline	Stops the configured service.
Monitor	Retrieves the current state of the configured service. It also verifies the user context, if applicable.

Agent state definitions

ONLINE	Indicates that the service being monitored is running.
OFFLINE	Indicates that the service being monitored is stopped.
UNKNOWN	Indicates the service operation is in a pending state, or that the agent cannot determine the state of the resource.

Agent resource type definition

```
type GenericService (
  static int IMF{} = { Mode=3, MonitorFreq=5, RegisterRetryLimit=3 }
  static i18nstr IMFRegList[] = { ServiceName }
  static i18nstr ArgList[] = { ServiceName, DelayAfterOnline,
    DelayAfterOffline, UserAccount, Password, Domain, service_arg,
    UseVirtualName, "LanmanResName:VirtualName" }
  i18nstr ServiceName
  int DelayAfterOnline = 10
  int DelayAfterOffline = 10
  i18nstr UserAccount
  str Password
  i18nstr Domain
  str service_arg[]
)
```

Agent attributes

The following table describes the GenericService agent required attribute.

Table A-8 GenericService agent required attribute

Required attribute	Description
ServiceName	Name of the service to be monitored. The service name can be the Service Display Name or the Service Key Name. Type and dimension: string-scalar

The following table describes the GenericService agent optional attributes.

Table A-9 GenericService agent optional attributes

Optional attributes	Description
DelayAfterOffline	Number of seconds the offline routine waits for the service to go offline. Default is 10 seconds. Type and dimension: integer-scalar
DelayAfterOnline	Number of seconds the online routine waits for the service to go online. Default is 10 seconds. Type and dimension: integer-scalar
Domain	The domain name to which the user specified in the UserAccount attribute belongs. If the UserAccount attribute is empty or contains a built-in service account, this attribute is ignored. Type and dimension: string-scalar
Password	The password of the user, in whose context, the service would be started. Encrypt the password using the VCSEncrypt utility. If the UserAccount attribute is empty or contains a built-in service account, this attribute is ignored. Type and dimension: string-scalar
service_arg	An array of arguments passed to the service. Type and dimension: string-vector

Table A-9 GenericService agent optional attributes *(continued)*

Optional attributes	Description
UserAccount	A valid user account in whose context the service will be monitored. The user name can be of the form username@domain.com or domain.com\username. If the startup type of the configured service is set to Automatic, then the user account you specify here must be the same as that specified in the Windows Service Control Manager (SCM). If you do not specify a value for this attribute, then the user account of the service in the SCM is ignored. To monitor service under built-in accounts, you must provide explicit values. For example: On Windows 2003: UserAccount='LocalSystem', 'Local Service', or 'Network Service'. Domain='NT Authority'. The 'NT Authority' domain is not applicable for the 'LocalSystem' account. Type and dimension: string-scalar

Process agent

The Process agent brings processes online, takes them offline, and monitors their status. You can specify different executables for each process routine. The processes are monitored in the context of the LocalSystem account by default. You can run a process with user privileges by specifying the user name, password, and domain.

This agent is represented by the Process resource type.

The Process agent supports intelligent resource monitoring and uses Intelligent Monitoring Framework (IMF) for resource state change notifications. The agent supports IMF-based monitoring only when the resource is in the online state.

Note: The Process agent does not use IMF notification for monitoring the program specified in the MonitorProgram attribute.

Agent functions

online Starts the process configured as the start program.

offline	Terminates the process, or starts the process configured as the stop program.
monitor	Verifies the status of the process, or starts the process configured as the monitor program.

State definitions

ONLINE	Indicates the process being monitored is running properly.
OFFLINE	Indicates the process being monitored is not running properly.
UNKNOWN	Indicates the agent cannot determine the status of the resource.

Resource type definition

```
type Process (
  static int IMF{} = { Mode=2, MonitorFreq=5, RegisterRetryLimit=3 }
  static i18nstr IMFRegList[] = { StartProgram, MonitorProgram }
  static i18nstr ArgList[] = { StartProgram, StopProgram,
    MonitorProgram, UserName, Password, Domain,
    MonitorProgramTimeout,InteractWithDesktop, CleanProgram,
    StartupDirectory, StopProgramTimeout, CleanProgramTimeout,
    "LanmanResName:VirtualName" }
  i18nstr StartProgram
  i18nstr StartupDirectory
  i18nstr StopProgram
  i18nstr CleanProgram
  i18nstr MonitorProgram
  i18nstr UserName
  str Password
  i18nstr Domain
  int MonitorProgramTimeout = 30
  boolean InteractWithDesktop = 0
  int StopProgramTimeout = 30
  int CleanProgramTimeout = 30
  str LanmanResName
)
```

Agent attributes

The following table describes the Process agent required attribute.

Table A-10 Process agent required attribute

Required attribute	Description
StartProgram	The process to be monitored by the agent. You must specify the complete path of the executable, its file extension, and command-line arguments, if any. If you define the start program as a batch file or a script to launch another program, you must specify the monitor program in the configuration file. If you define the start program as a script (a batch file, a Perl script, or a VBS script), the start program should be the program that interprets the script (cmd.exe, or perl.exe, or cscript.exe) and the script itself should be passed as an argument. Type and dimension: string-scalar

The following table describes the Process agent optional attributes.

Table A-11 Process agent optional attributes

Optional attributes	Description
CleanProgram	The full path of the clean process that is launched when the resource needs a forceful offline. If no value is specified for this attribute, for a clean operation the agent kills the process indicated by the StartProgram attribute. Type and dimension: string-scalar
CleanProgramTimeout	The maximum time, in seconds, that the agent must wait before killing the process specified in the CleanProgram attribute. This attribute is ignored if the clean program is not specified. The default value is 30 seconds. Type and dimension: integer-scalar
Domain	The domain in which the user specified by the attribute UserName exists. This attribute is ignored if the user name is not specified. Type and dimension: string-scalar

Table A-11 Process agent optional attributes *(continued)*

Optional attributes	Description
InteractWithDesktop	Defines whether the configured process interacts with the desktop. Enabling desktop interaction enables user intervention for the process. The value 1 indicates the process will interact with the desktop. The value 0 indicates it will not. Default is 0. Type and dimension: boolean-scalar
MonitorProgram	A program that monitors the process specified as the start program. You must specify the complete path of the executable, its file extension, and command-line arguments, if any. If you do not specify a value for this attribute, Disaster Recovery Orchestrator monitors the start program. However, if the start program is a batch file or a script to launch another program, you must specify a monitor program. The Process agent supports Intelligent Monitoring Framework (IMF). However, IMF is not supported for the MonitorProgram attribute. If a script or a batch file is specified for this attribute, the Process agent does not use IMF notification to monitor the program. The agent detects the faults only during the regular monitor function. Type and dimension: string-scalar **Note:** The monitor program is spawned every monitor cycle and must return before the program specified in MonitorProgram times out. The return values for the monitor program must conform to the Disaster Recovery Orchestrator conventions: 110 for ONLINE and 100 for OFFLINE. For exit values outside the range 100-110, the status is considered UNKNOWN.
MonitorProgramTimeout	The maximum wait time, in seconds, for the agent to receive a return value from the monitor routine. This attribute is ignored if the monitor program is not specified. Default is 30 seconds. Type and dimension: integer-scalar

Table A-11 Process agent optional attributes *(continued)*

Optional attributes	Description
Password	The encrypted password of the user specified by the UserName. Note that the password must be encrypted using the VCSEncrypt utility. This attribute is ignored if the user name is not specified. Type and dimension: string-scalar
StartupDirectory	The startup directory for the process indicated by the StartProgram attribute. Type and dimension: string-scalar
StopProgram	A program that stops the process specified as the start program. You must specify the complete path of the program, its file extension, and command-line arguments, if any. If you do not specify a value for this attribute, Disaster Recovery Orchestrator stops the start program. Type and dimension: string-scalar **Note:** If successful, the StopProgram returns a positive value. The Monitor routine is called after those many seconds, as returned by StopProgram. Also, while writing a stop program, make sure to stop all the processes launched by the start program. Type and Dimension: string-scalar
StopProgramTimeout	The maximum time, in seconds, that the agent must wait before killing the process specified in the StopProgram attribute. The default value is 30 seconds. Type and dimension: integer-scalar
UserName	The user name with whose privileges the configured process executes. User name can be of the form *username@domain.com* or *domain.com\username*. If a user name is not specified, the configured process runs in the context of the local system account. Type and dimension: string-scalar

Note: When defining the StartProgram, StopProgram, or MonitorProgram attributes, enclose the path of the executable file and its arguments in double quotes.

Appendix B

Synchronization of application data between on-premises and cloud systems

This appendix includes the following topics:

File replication in Disaster Recovery Orchestrator

Disaster Recovery Orchestrator uses file-level replication to ensure that application data from the primary site is replicated to the disaster recovery (DR) site. The file replication module is deployed as part of the Disaster Recovery Orchestrator installation.

The module is deployed on all the systems that participate in the DR solution for an application, which are:

- The on-premises system that hosts the application and the Disaster Recovery Orchestrator Client components
- The dedicated cloud virtual machine that hosts the Disaster Recovery Orchestrator Console components
- The cloud virtual machine that acts as the recovery site for the application, which also hosts the application and the Disaster Recovery Orchestrator Client components

File replication is a hardware-independent alternative to traditional array-based replication architectures. It enables cost-effective replication of data over IP networks, and is flexible. It allows the replication of a set of folders or volumes at the file system level, provided the disks are NTFS-formatted.

When you first configure an application for DR, the on-premises system is the primary site for replication and the Console host is the secondary site.

When the application or the on-premises system becomes unavailable, a recovery administrator intervenes to perform a Takeover operation. Disaster Recovery Orchestrator provisions an application virtual machine in the cloud to take over the application processing. The storage is detached from the Console host and attached to the cloud application host.The cloud application host becomes the primary site for replication, and the on-premises system—when it becomes available—becomes the secondary site.

The Takeover operation can also be used to perform a planned migration of the application to the cloud. In this case, the on-premises system is available, but the migration might be done for maintenance purposes. When the on-premises system is available, the Takeover operation triggers the 'migrate' file replication function. When the on-premises system is unavailable, the Takeover operation triggers the 'forced takeover' file replication function. The migrate function involves no data loss, whereas the forced takeover function may result in data loss. In the case of a forced takeover, the file replication mechanism recovers an older point-in-time data.

Later, when the application or the on-premises system becomes available, the recovery administrator performs a Failback operation. Disaster Recovery Orchestrator releases the cloud application host, detaches the storage, and attaches it back to the Console host. A differential synchronization operation is performed to make the most recent data available to the on-premises system. Then, the direction of replication is reversed again. The on-premises system becomes the primary site for replication and the Console host becomes the secondary site.

Disaster Recovery Orchestrator creates and maintains the following objects to manage replication:

- Replicated file groups
  See “Replicated file groups” on page 96.
- Replication links (RLINKs)
  See “Replication links” on page 97.
- Replication logs
  See “Replication logs (journal files)” on page 97.

For information about replication requirements to be met before you install or configure Disaster Recovery Orchestrator, see the *Symantec Disaster Recovery Orchestrator Deployment Guide*.

Replicated file groups

Disaster Recovery Orchestrator uses file replication to ensure that application data from the on-premises site is replicated to the cloud site for DR readiness. To manage replication, Disaster Recovery Orchestrator configures a set of application folders, which is called a replicated file group (RFG). An RFG is the unit of replication. An RFG can comprise of folders spread across multiple volumes.

The set of folders on a host that need to be replicated are grouped under an RFG and are referred to as the Primary RFG. To maintain consistency, the destination host to which the application data needs to be replicated also has a similar setup as that of the Primary RFG. This set of folders on the destination host is referred to as the Secondary RFG. The updates to the folders in an RFG on the Primary host are also sent to its Secondary hosts. When replication is active, access to the application data on the Secondary hosts is not allowed.

Replication logging information is stored in its own file, in a folder separate from the application data folders.

Note: A single RFG configuration may contain up to 16 different folders. Subfolders within these 16 folders are not counted individually.

Sample RFG configuration file

```
<?xml version="1.0"?>
  <x:vxfrcli xmlns:x="urn:vxfr">
  <command>
  <create>
  <rfg>
  <rfgname>RFG1</rfgname>
  <localip>10.217.52.124</localip>
  <srcfolder>E:\Source</srcfolder>
  <logfilelocation>E:\VFRLogs</logfilelocation>
  <logfilesize>100</logfilesize>
  </rfg>
  <secondary>
  <remoteip>10.217.52.127</remoteip>
  <targetfolder>E:\Target</targetfolder>
  <logfilelocation>E:\VFRLogs</logfilelocation>
  <logfilesize>100</logfilesize>
  </secondary>
  </create>
  </command>
  </x:vxfrcli>
```

Replication links

A replication link (RLINK) establishes the link between the primary RFG and secondary RFG.

The RLINK associated with the primary RFG controls replication settings, such as the following:

- mode of replication
- packet size used for replication
- latency or replication log protection protocol

Each RLINK associated with a Primary RFG represents one Secondary. Each RLINK associated with a Secondary RFG represents a Primary.

Note: RLINKs are not depicted in the Disaster Recovery Orchestrator GUI.

Replication logs (journal files)

Symantec Disaster Recovery Orchestrator uses a special file to store information about updates made to the application data folders on the primary site, so that it can be used to replicate those updates on the secondary site. This file stores information in the binary format, and is referred to as the journal file or the replication log.

All updates to the application data folders in the RFGPrimary are logged in the journal file at the primary site, before they are sent to the secondary site. Each update to the RFGPrimary generates two update write requests; one to the journal file and one to a data folder. Each RFG is associated with one journal file, and the file is required to maintain the consistency of the data between the hosts.

It is very important to plan the size and layout of the journal file appropriately. The appropriate size of the journal file can be derived from various criteria, however, the minimum size of the journal should not be less than 1 GB.

The replication log at the secondary site is only used to maintain data consistency while recovering from:

- a temporary failure in communication between the primary and secondary RFGs (or sites)
- a Primary or Secondary host failure

Note: The replication log is different from the replication service log. The replication service log, contains messages logged by the replication service that can be used to troubleshoot issues with the file replication.

Significance of replication log size

The size of the journal file is critical to the performance of replication. In the asynchronous mode of replication, due to network latency, some write operations may be pending on the journal file at the primary site. If the number of pending write operations exceed the number of updates that it can store, the journal file at the primary site may overflow.

When the journal file overflows for a particular secondary site, the RLINK corresponding to that secondary site is marked as STALE. Such an RLINK is considered outdated until a complete resynchronization with the primary site is performed. Because resynchronization is a time-consuming process and during this time the data on the secondary site cannot be used, it is important to avoid journal file overflows.

Thus, the journal file size needs to be large enough to not overflow during the following events:

- For asynchronous RLINKs during periods of peak usage when replication over the RLINK may fall far behind the rate at which application data is written
- During extended outages (network or secondary site)

Determining an appropriate replication log size

To determine the size of the journal file, evaluate each of the following constraints individually.

- The maximum expected downtime for Secondary nodes
- The maximum expected downtime for the network connection
- The method for synchronizing data folders at the secondary site with those from the primary site

Then, choose a value that is at least equal to the maximum so that all the constraints are satisfied.

Note: If the size of an existing journal file is not enough to meet new business requirements, you can resize the journal file.

If the application is shut down to perform data synchronization, the journal file is not used, and the method for determining its size is not important. Otherwise, this information should include the time required to copy the data over the network.

In the case of Secondary data volume failure if you are going to perform Secondary backup to avoid complete synchronization, the information needed includes the following:

- The frequency of Secondary backups.
- The maximum expected delay to detect and repair a failed Secondary data volume.
- The expected time to reload backups onto the repaired Secondary data volume.

Replication agents in Disaster Recovery Orchestrator

Disaster Recovery Orchestrator uses file replication to synchronize application data between the on-premises site and cloud site. Agents for file replication include type declarations and agent executables, which represent resource types.

The Symantec Disaster Recovery Orchestrator configuration wizard configures and manages replication configurations with the help of these agents.

The following agents are required to enable file-based replication:

- VFRRFG agent
- RFGPrimary agent

These agents are installed as a part of the Disaster Recovery Orchestrator installation process to manage the failover.

The following sections provide details about the file replication agents.

VFRRFG agent

The VFRRFG agent represents a replicated file group (RFG), which is the unit of replication. The agent enables the RFG to fail over from a primary site to a secondary site, thus making it highly available.

The VFRRFG resource represents the VFRRFG agent.

VFRRFG agent functions

Online	Enables data access to the folders in the RFG.
Offline	Disables data access to the folders in the RFG.

Monitor	Monitors the data access to the folders in the RFG and sets the resource status accordingly: ■ If data access is enabled, the resource appears ONLINE. ■ If data access is disabled, the resource appears OFFLINE

VFRRFG agent state definitions

ONLINE	Indicates that data access is enabled on the folders that are configured in the RFG.
OFFLINE	Indicates that data access is disabled on the folders that are configured in the RFG.
UNKNOWN	Indicates one or more of the following: ■ The agent cannot determine the status of the resource. ■ The resource corresponding to the RFGGuid attribute does not exist.

VFRRFG agent resource type definition

```
type VFRRFG (
  static i18nstr ArgList[] = { RFGGuid }
  str RFGGuid
)
```

VFRRFG agent attribute

Table B-1 Required VFRRFG agent attribute

Attribute	Description
Name: **RFGGuid** Type: **String** Dimension: **Scalar**	The unique identifier of the RFG that the agent manages.

Sample VFRRFG resource

A sample configuration of the VFRRFG resource is as follows:

```
VFRRFG Application_SG_RFG_VFRRes_Name (
  RFGGuid = "{3A5C473D-C8C8-471C-A366-1939F9113841}"
)
```

RFGPrimary agent

The RFGPrimary agent monitors the role of a replicated file group (RFG), which is the unit of replication. The agent enables the migrate or takeover operations for file replication in Disaster Recovery Orchestrator to make an application highly available across sites.

Whether the agent initiates the migrate or the takeover operation, depends on the following:

- State of the RFG
- Status of replication
- State of the recovery configuration

The RFGPrimary resource type represents the RFGPrimary agent.

RFGPrimary agent function

Online	Depending on network availability, converts the Secondary RFG to Primary by initiating either a migrate or a takeover operation.
Offline	Takes the resource offline.
Monitor	Determines the status of the RFG. If the RFG is Primary, it reports ONLINE and if the RFG is Secondary, it reports OFFLINE.

RFGPrimary agent state definitions

ONLINE	Indicates that the role of the corresponding RFG is Primary on the current system and Secondary on the other system in the recovery configuration.
OFFLINE	Indicates that the role of the corresponding RFG is not Primary on the current node. It does not necessarily mean that the RFG is Secondary. However, if the RFG is a Secondary and the RFGPrimary resource is brought online, then depending on the state of replication, the RFGPrimary agent performs a migrate or takeover operation. Thus, the agent monitors the role of the RFG and ensures that the RFG is Primary as long as the resource is online.
UNKNOWN	Indicates one or more of the following: ■ The agent cannot determine the status of the resource. ■ The corresponding RFG does not exist.

RFGPrimary agent resource type definition

```
type RFGPrimary (
  static i18nstr ArgList[] = { "RFGResourceName:RFGGuid" }
  str RFGResourceName
)
```

RFGPrimary agent attribute

Table B-2 Required RFGPrimary agent attribute

Attribute	Description
Name: **RFGResourceName** Type: **String** Dimension: **Scalar**	The name of the VFRRFG resource in the replication group on which the application configuration depends.

Sample RFGPrimary resource

A sample configuration of the RFGPrimary resource is as follows:

```
RFGPrimary Application_SG_RFG_Res_Name (
RFGResourceName = Application_SG_RFG_VFRRes_Name
)
```

Appendix C

Troubleshooting

This appendix includes the following topics:

Disaster Recovery Orchestrator logging

Disaster Recovery Orchestrator provides the following logging information.

Installation logs

Disaster Recovery Orchestrator installer logs contain details about the installation tasks and the overall progress status. These logs are useful for identifying installation-related issues.

The installer creates the log directory as soon as you launch the wizard. The log files are located at:

`%AllUsersProfile%\Veritas\VPI\log\`

`%AllUsersProfile%` expands to `C:\ProgramData`.

Console logs

The Disaster Recovery Orchestrator Console logs are located at:

`%AllUsersProfile%\symantec\draasconsole\Logs`

`%AllUsersProfile%` expands to `C:\ProgramData`.

The Console logs are written to the `azuredraas.log` file.

The components of the Console logs are as follows:

- Timestamp
  The date and time the message was generated
- Duration
  The number of milliseconds elapsed between the construction of the layout and the creation of the logging event
- Thread
  The name of the thread that generated this logging event
- Priority
  Levels in the increasing order of priority: TRACE, DEBUG, INFO, WARN, ERROR, and FATAL
- Object
  The logger object, which the fully qualified class name of the caller that issues the logging request
- Message
  The actual message that was generated by the thread

Additionally, the Disaster Recovery Orchestrator UI components create logs that are available only for the duration of their existence. For example, the Console UI logs are available only as long as you are signed in and the session is active. These logs are lost after the session ends. Similarly, the Disaster Recovery OrchestratorConfiguration Wizard creates a log file that is available from within the wizard. This information is lost when you exit the wizard.

Agent logs

The agent logs are located at:

```
C:\Program Files\Veritas\cluster server\log
```

The components of the agent logs are as follows:

- Timestamp
  The date and time the message was generated
- Mnemonic
  The string ID that represents the product, for example, SDRO
- Severity
  Levels in the increasing order of severity: INFO, NOTICE, WARNING, ERROR, and CRITICAL

- UMI
  A unique message ID
- Message
  The actual message generated by the agent

Collecting Disaster Recovery Orchestrator logs

Disaster Recovery Orchestrator provides the `hagetcf` utility, which you can use to collect logs. This utility retrieves detailed diagnostic information about your application monitoring and recovery configurations. You can use this information to troubleshoot configuration-related issues. You can also share these logs with Symantec Technical Support for further troubleshooting.

The `hagetcf` utility is available at the following locations:

- On the Console host (Azure virtual machine where Disaster Recovery Orchestrator Console is installed):
  `InstallDir\draasconsole\bin`
  Here, *InstallDir* is the Disaster Recovery Orchestrator Console installation directory, typically, `C:\Program Files\Veritas`.
- On a system where Disaster Recovery Orchestrator Client is installed:
  `%vcs_home%\bin`
  The `%vcs_home%` environment variable points to the product home directory, typically, `C:\Program Files\Veritas\Cluster Server`.

To collect Disaster Recovery Orchestrator logs

1. On the Console host, navigate to the location where the utility is installed.
   This step is not required on Disaster Recovery Orchestrator Client systems.
2. Collect the logs using the following command:
   `hagetcf [-Option]`
   You can limit the diagnostic information to specific components using the various available options.
   Use the `-?` or `-help` option to view the command's usage information.

Note: If you do not specify any options, the command retrieves diagnostic information with the options: `-app`, `-sys`, `-ha`, `-log`, `-lock`, `-conf`, `-state`,`-islog`, and `-trigger`. On a Console host, it also includes the logs for the Disaster Recovery Orchestrator Console component.

By default, `hagetcf` writes the output to the following locations:

- On the Console host:
  `%AllUsersProfile%\Symantec\hagetcf\`*`mmdd_hhmm`*
  The `%AllUsersProfile%` environment variable points to the common program data location, typically, `C:\ProgramData`.
- On Disaster Recovery Orchestrator Client systems:
  %vcs_home%\hagetcf*mmdd_hhmm*

The *mmdd_hhmm* folder name indicates the date and time when the logs were collected, for example: `C:\Program Files\Veritas\Cluster Server\hagetcf\0428_1520`. The folder contains several subfolders and log files, which represent various components.

Disaster Recovery Orchestrator UI issues and solutions

The following sections describe some of the issues that you might encounter while working with the Disaster Recovery Orchestrator UI, and provide solutions to work around those issues.

Disaster Recovery Orchestrator Console views are not displayed correctly

Disaster Recovery Orchestrator Console views use Adobe Flash components. Therefore, Flash Player is required for those views to be displayed correctly. If the views are not displayed correctly:

- On a Windows Server 2008 R2 system, check whether the latest Adobe Flash Player plugin is installed and enabled.
- On a Windows Server 2012 system, check whether Desktop Experience feature is enabled.

Enabling Adobe Flash Player on Windows Server 2012

Internet Explorer 10 is available by default with Windows Server 2012, which includes Adobe Flash Player as a platform feature. However, the plugin is enabled only if you enable the Desktop Experience feature.

For more information, see the Microsoft article:

Internet Explorer 10 FAQ for IT Pros

To enable Desktop Experience on Windows Server 2012

1. Open Server Manager and click **Add Roles and Features**.
2. When the Add Roles and Features Wizard appears, specify the appropriate values on the **Installation Type**, **Server Selection**, and **Server Roles** pages.
3. On the Features page, expand **User Interfaces and Infrastructure** and select **Desktop Experience**.
4. On the Confirmation page, select **Restart the destination server automatically if required** and click **Install**.

Note: If you don't select this checkbox, you'll need to restart the server manually before you can use the Adobe Flash Player plugin.

Application monitoring configuration issues and solutions

The following sections describe some of the issues that you might encounter while configuring applications for monitoring, and provide solutions to work around those issues.

Health View fails to open on a system where Disaster Recovery Orchestrator Client is installed

If you encounter this issue, take the following actions:

- Make sure that the Symantec Storage Foundation Messaging Service (`xprtld`) service is running on the system.
- If the `xprtld` service is running, check your domain controller configuration using the following command:

  `nltest /DSGETDC:`*`DomainName`*

  Make sure that the following output string points to the local site:

  `Our Site Name:` *`LocalSiteName`*

  If it points to a non-local site, fix the domain controller configuration as per Symantec recommendations.

  For information about the domain confirguration recommendations, refer to the *Symantec Disaster Recovery Orchestrator Deployment Guide*.

Application monitoring configuration fails to come online after the server restarts or after application downtime

By default, the application monitoring configuration does not start automatically on a system after it restarts or after application downtime.

Workaround

To bring the configuration online, perform the following steps.

1. Log on to the system that hosts the application.
2. Launch the Health View using the following URL:

   `https://SystemName:5634/vcs/admin/application_health.html`
3. When prompted, provide the appropriate login credentials.
4. Click the **Start Application** link.

On the Disaster Recovery Orchestrator Console Dashboard or Applications view, verify that the application configuration appears online.

Disaster recovery configuration issues and solutions

The following sections describe some of the issues that you might encounter while configuring applications for recovery, and provide solutions to work around those issues.

Application cannot be configured for recovery if the maximum permitted disks are attached to Console host

When you launch the Disaster Recovery Orchestrator Configuration Wizard, you may encounter the following error:

```
Disaster Recovery Orchestrator Console has reached at the maximum
permitted data disk count. It is required to remove one or more
configured applications to proceed with this configuration.
```

When this message appears, you cannot proceed with creating an application recovery configuration.

Workaround

The number of disks that can be attached to a cloud virtual machine depends on the size of the virtual machine. For information about the virtual machine sizes available in Azure, refer to the Microsoft article:

http://msdn.microsoft.com/library/azure/dn197896.aspx

If you want to configure another application for recovery, perform one of the following activities:

- If possible, increase the size of the Console host. Refer to the following procedure:
  To increase the size of the Console host

  Note: Increasing the virtual machine size affects the Microsoft Azure subscription cost.

- If you do not want to increase the size of the Console host, remove an existing application recovery configuration. The number of disks attached to the Console host is reduced, and you can try configuring the new application again.

To increase the size of the Console host

1. Wait for any ongoing Firedrill, Takeover, or Failback operation to complete.
2. Pause the replication activity for each application recovery configuration.

   Run the following commands from the system where the application is currently online:

   - `vxfradmin -viewconfig`
     This command displays the replicated file group (RFG) name for the application recovery configuration.
   - `vxfradmin -pauserep` *`RFGName`*
   - Replace the *RFGName* variable with the correct value to pause the replication.
3. Sign in to the Azure Management Portal and increase the size of the virtual machine. For details, refer to the Microsoft article:

   http://msdn.microsoft.com/en-us/library/dn168976%28v=nav.70%29.aspx

   The virtual machine restarts.
4. Resume the replication activity for each application recovery configuration.

   Run the following commands from the system where the application is currently online:

   `vxfradmin -resumerep` *`RFGName`*

   Replace the *RFGName* variable with the RFG name that you used to pause the replication.

Creation of a file replication configuration (RFG) fails due to connection issues

When configuring an application for disaster recovery (DR) in the cloud using the Disaster Recovery Orchestrator Configuration Wizard, you might encounter the following message:

```
Failed to connect remote instance. (e0000029)
```

This message is visible in the wizard log as well as next to the failed task that is displayed on the UI.

This issue may occur due to one of the following reasons:

- One of the systems that are required to establish the replication configuration is not accessible.
- The port that the replication service requires is not added as an exception on the firewall.
- The DCOM settings are incorrect.

Workaround

Perform the following tasks to resolve the issues mentioned previously:

- For each system associated with the application recovery configuration, check whether the system name resolves to the correct IP address. Ping the system to check whether it is accessible.
- Check whether port 14159 is added to the exceptions on the firewall. If not, add the port as an exception so that it can be used for configuring the replication.
- By default, DCOM is enabled on a Windows Server system. If it has been disabled for some reason, make sure to enable it so that you can proceeding with the DR configuration.

  To enable DCOM on a system

  Check whether the appropriate DCOM permissions are set for all the systems associated with the application recovery configuration. If not, add the permissions for the required systems.

  To change DCOM settings

  If the DCOM settings are correct and you still encounter this issue, check the user configuration. Add the same user as the logon user for the file replication service on all the systems associated with the application recovery configuration. This user must also have local administrator privileges on all the systems.

To enable DCOM on a system

1. Log on to the system with local administrator privileges.
2. Run the `dcomcnfg` command to open the Component Services window.

3 Expand **Component Services > Computers**.

4 Right-click **My Computer** and select **Properties** from the context menu.

5 On the My Computer Properties dialog box, open the Default Properties tab.

6 If the **Enabled Distributed COM on this computer** check box is not already selected, select it.

7 If you enabled DCOM, click **OK** to save the change. Then, restart the computer for this change to take effect.

If DCOM was already enabled, click **Cancel** to close the dialog box.

To change DCOM settings

1 Access the My Computer Properties dialog box in the Component Services window as described in the previous procedure.

2 Open the COM Security tab, and click the **Edit Limits...** button in the Access Permissions section.

On the Access Permission dialog box, the systems associated with the application recovery configuration should be listed. Select each system to check whether the remote access permission is granted to it.

If a system associated with the application recovery configuration is not listed, click the **Add...** button. The detailed steps are described in the following section:

To add permissions for specific systems

Select the system that you added, enable **Remote Access**, and click **OK**.

3 Similarly, check whether the appropriate Launch and Activation Permissions are granted to each system associated with the application recovery configuration.

If not, add the appropriate systems, and set the **Remote Launch** and **Remote Activation**permissions.

4 Click **OK** to close the My Computer Properties dialog box.

5 In the Component Services window, expand **My Computer > DCOM Config**.

6 Right-click **{807ADEFE-5D10-46FE-A8EA-261545836C2E}** and select **Properties** from the context menu.

7 On the {807ADEFE-5D10-46FE-A8EA-261545836C2E} Properties dialog box, open the Security tab.

8 In the Launch and Activation Permission section, select the **Customize** option, and click the **Edit...** button.

Check whether the Remote Launch and the Remote Activation permissions are granted to each system associated with the application recovery configuration. If not, add the appropriate systems, and set the permissions like you did earlier for My Computer.

9 In the Access Permissions section, select the **Customize** option, and click the **Edit...** button.

Check whether the remote access permission is granted to each system associated with the application recovery configuration. If not, add the appropriate systems, and set the permission like you did earlier for My Computer.

10 Click **OK** to close the {807ADEFE-5D10-46FE-A8EA-261545836C2E} Properties dialog box.

To add permissions for specific systems

Note: All the systems that participate in the DR solution for the application should be listed in the appropriate permissions dialog box. When checking the DCOM settings, if you find that a system is not listed, perform this procedure.

---

1 On the Select Users, Computers, Service Accounts, or Groups dialog box, click the **Object Types...** button.

2 On the Object Types dialog box, select **Computers** and click **OK**.

3 On the Select Users, Computers, Service Accounts, or Groups dialog box, enter the correct system name and click **OK**.

- On the Console host, select the on-premises application host and the corresponding cloud application host.
- On the on-premises application host, select the cloud application host and the Console host.
- On the cloud application host, select the on-premises application host and the Console host.

Finalizing an application recovery configuration fails if 26 or more volumes are attached to Console host

When finalizing an application recovery configuration, the last task is to attach the storage to the Console host. During this task, you might encounter an issue, which is indicated by the following message in the log:

```
SFR Error <error code>: Object not found
```

This issue occurs if 26 or more volumes are already attached to the Console host. The additional volumes cannot be automatically mounted, because no more drive letters are available.

Workaround

Create a folder mount for each volume that was detached from the cloud application host and must now be attached to the Console host. Retry the finalize operation.

For information about finalizing application recovery configurations:

See "Finalizing the application recovery configuration" on page 30.

For information about the supported folder mount configurations, refer to the *Symantec Disaster Recovery Orchestrator Deployment Guide*.

Finalizing an application recovery configuration fails or the recovery operations fail when changing the replication settings on a system

The replication settings on the Console host and the cloud application host are changed during the following operations:

- Finalize application recovery configuration
- Takeover
- Failback

While performing these operations, the following task may fail:

```
Change replication settings on 'System'
```

The *System* variable represents the cloud virtual machine on which this task is performed.

Workaround

If you encounter this issue, verify the following:

- The volumes are mounted with a drive letter on the cloud virtual machine that is the replication target.
- No network connection issues exist between the following systems:
  - While finalizing the application recovery configuration and during Failback, the on-premises application host and the Console host
  - During Takeover, the on-premises application host and the cloud application host

DR configuration or recovery operation fails due to mismatched virtual machine name values

If the Name and the Host Name values of an Azure virtual machine are different, you might encounter the following issues:

- In case of the Console host, Disaster Recovery Orchestrator fails to finalize the application recovery configuration step fails. For an existing configuration, it fails to perform the failback operation.
- In case of the cloud application host, Disaster Recovery Orchestrator fails to perform the takeover operation.

The Name value is provided in the Virtual Machine Name field when the Azure virtual machine is created. The Host Name is derived from the Name when the virtual machine is created, but it can be changed later. Check these values on the virtual machine details page of the Azure management portal.

Workaround

If these values are different for an Azure virtual machine, log on to the virtual machine and change the Host Name value. Edit the **Computer name** field on the Computer Name/Domain Changes dialog box, and restart the virtual machine to reflect the change.

Verify that the Name and the Host Name values of the virtual machine now match. Then, perform the following tasks:

- Retry the failed task from the Disaster Recovery Orchestrator Console UI.
- If the task still fails, remove the application recovery configuration and run the Disaster Recovery Orchestrator Configuration Wizard again.
  See "Removing the recovery configuration of an application" on page 57.
  See "Configuring an application for disaster recovery" on page 27.

Takeover of an application fails due to insufficient user privileges

Even though an application is successfully configured for recovery in the cloud, a takeover operation on the application might fail. The task of bringing the application online on the cloud application host fails, which is reflected on the Takeover view and in the Disaster Recovery Orchestrator log. The underlying issue is that the Lanman resource fails to come online, and the following entry can be found in the Lanman log:

```
Lanman:lanmanResouceName:online:Failed to update DNS entry
(error_type:2, error_code:0x0000232D)
```

The following entry appears in the Windows Event Viewer:

```
HadHelper warning message: Internal error (RegCreateKeyEx() failed)
(status=0x00000005)
```

The Health View on the cloud application host depicts that the Lanman resource is unable to come online and reports its state as Faulted.

Workaround

If you encounter this issue, check whether the user whose credentials you provided on the Virtual Computer Name panel has the following privileges:

- DNS administrator
- Local administrator on the on-premises application host and the corresponding cloud application host

The application monitoring helper service runs under this user context and updates the DNS with the virtual computer name. If the user does not have the required privileges, add the user to the appropriate groups and set the privileges. Then, relaunch the Disaster Recovery Orchestrator Configuration Wizard.

Application data synchronization issues and solutions

The following sections list some of the issues that you might encounter with the file replication configurations, and provide solutions to work around those issues.

Synchronization of application data fails

The synchronization of the application data between the on-premises and the cloud sites may fail due to various reasons, some of which are:

- The required ports are not open for communication.
- The required users are not configured or they do not have the correct privileges.

Workaround

To resolve this issue, check the following configuration settings and perform the corresponding tasks:

- If a firewall is enabled on a system that hosts the file replication service, add exceptions to allow traffic across the firewall. These exceptions should include the default file replication port (14159) and any user-configured ports.
- Make sure that the required users are configured on all the systems associated with the application recovery configuration. If not, identify or create the users with the appropriate permissions.

For information about the network, security, and file replication requirements, see the *Symantec Disaster Recovery Orchestrator Deployment Guide*.

Data replication stops unexpectedly if the journal file or the in-memory log queue is full

Data replication might stop if the file replication configuration (RFG) goes into the Stopped state.

Some of the reasons in which this state occurs are:

- The journal file size is insufficient or the file is full
- The in-memory log queue size is insufficient or the queue is full

Workaround

When the replication stops, the memory is released and the journal file is flushed. To proceed, sign in to the Console UI, select the affected application from the Applications view, and click **Start Replication**.

Optionally, you can perform the following tasks:

- Increase the journal file size for the application recovery configuration.
  To do so, use the Settings page of each application on the Console UI.
  See "Changing the disaster recovery settings of an application" on page 54.
  However, you may encounter performance issues even if this size is larger than the requirement. In this case, check the in-memory log queue size.
- By default, the in-memory log queue size is set to 2 GB. If the replication stops frequently because this queue gets full, you might want to increase its size.
  To do so, add the `DefaultLogBufferMaxSize` registry key and set it to an appropriate value as described in the following procedure:
  To increase the in-memory log queue size for replication

Note: The file replication service reads this value only when its driver is loaded. Therefore, you need to restart the service to implement this solution.

Warning: Before you reload the replication driver and restart the replication service, you must stop the ongoing replication. After starting the service, you need to start the replication again. The initial synchronization is performed when you start the replication, which may take a long time depending on the data that needs to be copied. During this time, no other operations can be performed on the application recovery configuration.

To increase the in-memory log queue size for replication

1. Create the following registry key:

   ```
   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\VxRepREG_DWORD
   "DefaultLogBufferMaxSize" = SizeInBytes
   ```

   Specify *SizeInBytes* to be a value that is suitable to your replication requirements.

   Note: Back up the existing registry keys before you modify them.

2. On the Applications view of the Console UI, select the application, and click **Stop Replication**.
3. Perform the following steps sequentially on the replication primary (the on-premises application host or the cloud application host):
   - Run the following command to stop the replication service:
     ```
     net stop vxrepservice
     ```
   - Run the following command to unload the replication driver:
     ```
     fltmc unload vxrep
     ```
   - Run the following command to load the replication drivers:
     ```
     fltmc load vxrep
     ```
   - Run the following command to start the replication service:
     ```
     net start vxrepservice
     ```
4. On the Applications view of the Console UI, select the application, and click **Start Replication**.

Pausing the replication fails when performing a takeover or failback operation on an application

There could be several reasons due to which Disaster Recovery Orchestrator fails to perform the pause replication task as part of the takeover or failback operation.

One of the scenarios in which you could encounter this issue is as follows. You stop or start the replication for an application from the Disaster Recovery Orchestrator Console UI. Then, before the changed replication status is reflected on the Applications view, you perform a takeover or failback on the same application.

Note: You can perform a fire drill at any point, as long as the application is available at the on-premises site. This issue does not occur with the fire drill operation.

In this scenario, you can identify the issue from the following locations:

- On the Takeover view or the Failback view, the 'Pause replication' task for that application is reported as failed.
- On the Applications view, the Operation column for that application reports that the operation has failed.

Workaround

In case of the previously scenario mentioned, you need to perform the following tasks to restore the correct replication status and complete the intended operation.

1. On the Takeover view or the Failback view of the Disaster Recovery Orchestrator Console UI, select the appropriate application and click **Terminate** on the command bar.
2. On the appropriate Disaster Recovery Orchestrator Client system (the replication primary), open the Health View and click **Start Application**.
3. On the Applications view of the Disaster Recovery Orchestrator Console UI, wait for the replication status of the appropriate application to be updated (approximately 60 seconds).

   Then, select the application and click **Start Replication** on the command bar.

   After the replication status is updated (approximately 60 seconds), click **Takeover** or **Failback** on the command bar.

Index

S

T

U

V

W