Symantec™ Disaster Recovery Orchestrator Configuration Guide for Microsoft SQL Server 2012

Amazon Web Services

6.1.1

November 2014

Symantec™ Disaster Recovery Orchestrator Configuration Guide for Microsoft SQL Server 2012

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: 6.1.1

Document version: 6.1.1 Rev 0

Legal Notice

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.

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_apj@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

Introducing the Disaster Recovery Orchestrator agent for Microsoft SQL Server

This chapter includes the following topics:

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 recovery 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.

About the agent functions and attributes

Every agent has a collection of attributes and performs a definite set of functions.

An agent uses the values of its attributes to configure the corresponding application component to function in a specific way. By modifying attribute values you can change the way in which Disaster Recovery Orchestrator agent manages the component.

For example, the IP agent monitors an IP address. The specific address to be monitored is identified by value of the `Address` attribute.

Depending on the category to which an agent belongs, an agent performs either or all of the following functions:

Online	Brings the configured component online
Offline	Takes the configured component offline
Monitor	Verifies whether the configured component is online

As part of the Monitor function, an agent reports the following states:

ONLINE	Indicates that the configured component is online
OFFLINE	Indicates that the configured component or application has faulted
UNKNOWN	Indicates that the agent encountered issues while monitoring the configured component

About the Disaster Recovery Orchestrator agents for SQL Server 2012

The Disaster Recovery Orchestrator agents for SQL Server provide monitoring support for SQL Server 2012 (including SP1).

The agents monitor the SQL Server instances and the associated services on the system where the application is installed.

The Disaster Recovery Orchestrator agents for SQL Server are:

- SQL Server 2012 Database Engine agent
  The agent monitors the status of the SQL Server Database Engine service. If the service is not running, the agent declares the corresponding resource as Offline.
  See "About the Disaster Recovery Orchestrator agent for SQL Server Database Engine" on page 12.
- SQL Server 2012 FILESTREAM agent
  The agent monitors the Windows FILESTREAM configuration settings for the SQL Server instance.
  See "About the Disaster Recovery Orchestrator agent for SQL Server FILESTREAM" on page 16.
- GenericService agent
  The agent monitors the SQL Server 2012 Agent service and Analysis service.
  If the service is not running, the agent declares the corresponding resource as Offline.

About the Disaster Recovery Orchestrator agent for SQL Server Database Engine

The Disaster Recovery Orchestrator agent for SQL Server Database Engine agent monitors the Database Engine service. The agent brings the service online, monitors the status, and takes it offline. If the service is not running, the agent declares its state as offline.

If detail monitoring is configured, the agent checks the health of the configured SQL Server databases or executes a monitoring script. If detail monitoring is successful, the agent declares the application as available.

The agent for SQL Server 2012 is configured as a resource of the `SQLServer` type.

Agent functions

Online	Brings the SQL Server service online.
Offline	Takes the SQL Server service offline.
Monitor	Monitors the status of SQL Server services. If detail monitoring is configured, then depending on the configuration, the agent performs a database health check or executes a monitoring script.
Clean	Forcibly stops the SQL Server service.

Resource type definition

```
type SQLServer (
  static int IMF{} = { Mode=3, MonitorFreq=5, RegisterRetryLimit=3 }
  static i18nstr IMFRegList[] = { Instance }
  static i18nstr ArgList[] = { Instance, "LanmanResName:VirtualName",
    DetailMonitorInterval, SQLOnlineTimeout, SQLOfflineTimeout,
    SQLDetailMonitorTimeout, Username, Domain, Password, DBList,
    "IPResName:Address", SQLFile, FaultOnDMFailure }
  str Instance
  str LanmanResName
  str IPResName
  int DetailMonitorInterval = 0
  int SQLOnlineTimeout = 90
  int SQLOfflineTimeout = 90
  int SQLDetailMonitorTimeout = 30
  i18nstr Username
  i18nstr Domain
  str Password
  i18nstr SQLFile
  i18nstr DBList[]
  boolean FaultOnDMFailure = 1
)
```

Attribute descriptions

Table 1-1 Disaster Recovery Orchestrator agent for SQL Server Database Engine—required attributes

Attribute	Description
Name: **Instance** Type: **String** Dimension: **Scalar**	Name of the SQL Server instance to monitor. If the value of this attribute is empty, the agent monitors the default SQL Server instance (`MSSQLSERVER`).
Name: **LanmanResName** Type: **String** Dimension: **Scalar**	Lanman resource name on which the SQLServer resource depends.

Table 1-2 Disaster Recovery Orchestrator agent for SQL Server Database Engine—optional attributes

Attribute	Description
Name: **SQLOnlineTimeout** Type: **Integer** Dimension: **Scalar**	Number of seconds that may elapse before the Online function aborts. Default value: 90
Name: **SQLOfflineTimeout** Type: **Integer** Dimension: **Scalar**	Number of seconds that may elapse before the Offline function aborts. Default value: 90

Table 1-2 Disaster Recovery Orchestrator agent for SQL Server Database Engine—optional attributes *(continued)*

Attribute	Description
Name: **DetailMonitorInterval** Type: **Integer** Dimension: **Scalar**	Defines the level of detail monitoring that the agent performs for SQL Server. The value 0 (zero) indicates that the agent performs only the basic monitoring of the instance service. A non-zero value indicates the number of online monitor cycles that the agent must wait before performing detail monitoring. Symantec recommends that you set this value to an integer between 1 and 12. The value 1 would make the agent perform detail monitoring at each monitor cycle. The value 2 would make the agent perform detail monitoring at every other monitor cycle. This interpretation may be extended to other values. If this attribute is set to a non-zero value, then the following attributes must be assigned the appropriate values: ■ For script-based monitoring: SQLFile, Username, Password, Domain, and SQLDetailMonitorTimeOut ■ For databases list-based monitoring: Username, Password, Domain, and DBList Default value: 0 **Note:** The Application Monitoring Configuration wizard sets the value of this attribute to 5.
Name: **FaultOnDMFailure** Type: **Boolean** Dimension: **Scalar**	Defines whether the agent faults the SQL Server resource when the detail monitoring fails. ■ If the value is set to True, the agent faults the configured SQL Server resource when the detail monitoring fails. In this case, the SQL Server resource may go into OFFLINE state. ■ If the value is set to False, the agent does not fault the SQL Server resource when the detail monitoring fails. In this case, the SQL Server resource may go into the UNKOWN state. Default value: True
Name: **SQLDetailMonitorTimeout** Type: **Integer** Dimension: **Scalar**	Number of seconds that may elapse before the detail monitoring routine aborts. Default value: 30
Name: **Username** Type: **String** Dimension: **Scalar**	User account in whose context the detail monitoring is performed. If the DetailMonitorInterval attribute is set to a non-zero value, the value of this attribute must not be empty.

Table 1-2 Disaster Recovery Orchestrator agent for SQL Server Database Engine—optional attributes *(continued)*

Attribute	Description
Name: **Domain** Type: **String** Dimension: **Scalar**	Domain of the user account specified in the Username attribute.
Name: **Password** Type: **String** Dimension: **Scalar**	Password for the user account specified in the Username attribute. If the DetailMonitorInterval attribute is set to a non-zero value, the value of this attribute must not be empty. This password is encrypted.
Name: **SQLFile** Type: **String** Dimension: **Scalar**	Location of the SQL Server file executed during a monitor cycle. If the DetailMonitorInterval attribute is set to a non-zero value, then either the script-based detail monitoring or the database list-based detail monitoring must be configured.
Name: **DBList** Type: **String** Dimension: **Vector**	List of databases for which the agent must perform detail monitoring. If the DetailMonitorInterval attribute is set to a non-zero value, then either the script-based detail monitoring or the database list-based detail monitoring must be configured. If both the SQLFile attribute and the DBList attribute are configured, then the DBList attribute takes precedence.

About the Disaster Recovery Orchestrator agent for SQL Server FILESTREAM

The Disaster Recovery Orchestrator agent for SQL Server FILESTREAM enables FILESTREAM storage for the specified SQL Server instance, monitors its status, and disables it.

FILESTREAM enables SQL Server-based applications to store unstructured data, such as documents and images, on the file system.

The FILESTREAM resource type represents this agent.

Agent functions

Online	Enables FILESTREAM on the system.
Offline	Disables FILESTREAM on the system.

Monitor	Monitors FILESTREAM status on the system. If the agent is unable to query the status of FILESTREAM or if FILESTREAM is disabled on the system, the FILESTREAM resource in the application monitoring configuration faults.
Clean	Cleans up the resource state after the resource fails to come online, fails to go offline, or fails to be detected as online even though it is in the ONLINE state.

Resource type definition

```
type SQLFilestream (
static i18nstr ArgList[] = { InstanceName }
i18nstr InstanceName
)
```

Agent attributes

Table 1-3 Disaster Recovery Orchestrator agent for SQL Server FILESTREAM—required attribute

Attribute	Description
Name: **InstanceName** Type: **String** Dimension: **Scalar**	Name of the SQL Server instance for which FILESTREAM is configured. If the value of this attribute is empty, the agent monitors the default SQL Server instance (`MSSQLSERVER`).

How the Disaster Recovery Orchestrator agent monitors SQL Server

The Disaster Recovery Orchestrator agents for SQL Server monitor the configured resources, determine the status of these resources, bring them online, and take them offline. The agents detect an application failure if the configured SQL Server instance or associated services become unavailable. The agents try to start the application services for a configurable number of attempts. If the application services fail to start, the agents consider this as an application failure.

The agent for SQL Server provides the following levels of monitoring support:

- Basic monitoring
  In the basic level monitoring, the agent monitors and determines if the configured resources are online and the corresponding SQL Server instance and associated services are running.
- Detail monitoring

In detail monitoring, the agent runs a user-defined SQL script or connects to the configured databases to verify the status of SQL Server instance.

- In the case of script-based monitoring, it detects an application failure if the script execution fails.
- In the case of DB-list based monitoring, the agent attempts to connect to the configured DBs.

The following exit codes indicate the status of the script execution or the status of the connection to the configured DBs:

Exit code	Status
0	Success; the agent reports that the SQL Server resource is in the ONLINE state.
Any other	The value of the FaultOnDMFailure attribute determines the state of the resource as follows: ■ If the value is set to `True`, the agent reports that the SQL Server resource is in the OFFLINE state. This implies that the resource has faulted. ■ If the value is set to `False`, the agent reports that the SQL Server resource is in the UNKNOWN state.

Configuring disaster recovery

This chapter includes the following topics:

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:

- The appropriate IAM users and permissions must be configured so that the Disaster Recovery Orchestrator components can access the required AWS resources.
- The ports required by Disaster Recovery Orchestrator must be open for communication on the Console host and the Client hosts.
- The reverse DNS and PTR records must be configured so that the DNS can map an IP address to a domain name.

Note: If you do not set up the reverse lookup zone, the application recovery configuration may fail to come online.

- 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 Configuration wizard prompts you to do so. You can launch the Application Monitoring Configuration wizard from within the Disaster Recovery Configuration wizard. After the application monitoring configurations are in place on both the systems, you can proceed to configure the application for recovery.
- The latest Adobe Flash Player plug-in must be available for the browser that you use to access the Console UI. Flash Player must be enabled for use on the systems that run Windows Server 2012 or 2012 R2.

Configuration

- Ensure that User Access Control (UAC) is disabled on all the systems that participate in the DR solution.
- Ensure that the date and time settings on all the systems that participate in the DR solution are in sync with the domain controller.
- If you use the same AMI to launch multiple EC2 instances, make sure that the Computer Name of the Windows instance is unique in the VPC.
- Ensure that the appropriate users are configured on the Privilege Settings view of the Console UI.
- The following SQL Server configuration items on the cloud application hosts must match those on the corresponding on-premises application hosts:
  - Installation directory, name, and ID of each instance
  - Drive letters of the data volumes
  - Installation directory and name of each user-defined database
  - Users
- The same user must be configured as the SQL Server administrator on the on-premises application host and the corresponding cloud application host.
- The SQL Server client tools must be installed on the on-premises application host or the cloud application host. If they are not installed, ensure that the following patches are applied on these systems in the following order:
  - http://www.microsoft.com/en-us/download/confirmation.aspx?id=36434
  - http://www.microsoft.com/en-us/download/confirmation.aspx?id=36433

These patches are required for the ODBC driver and the `sqlcmd` utility commands to work.

- Make sure that the user who launches the Disaster Recovery Configuration wizard has permissions for the serveradmin and sysadmin roles on SQL Server 2012.
  For more information about the SQL Server roles, see the Microsoft documentation.

Storage and replication

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

- No additional volumes, other than the root device volume itself or the volumes of other application recovery configurations, must be attached to the Console host.
- For the duration that an Amazon EC2 instance acts as the Console host, a volume must not be manually attached to or detached from it. Disaster Recovery Orchestrator must manage the addition or removal of any storage on this instance.
- A volume must not be part of a Windows Server storage pool. If you use a volume that is created in a storage pool, the takeover and failback operations will eventually fail.
- The Windows automount feature must be enabled on the EC2 instances. 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 to configure 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 to a larger size, for example, 10 GB. If you define a small journal file size and your application data I/O rate is high, the journal file may become full quickly. When the journal file is full, the replication stops. Maintaining a large-sized journal file helps avoid such issues and also provides a better replication 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 volume 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

Use the Disaster Recovery Configuration wizard to configure an application for recovery or to migrate it to the cloud.

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.

     The **System Name** field lists the names of the on-premises systems that satisfy the following criteria:
     - The system belongs to the same domain as the user who launches this wizard.
     - The user who launches this wizard is configured as a recovery administrator on the system.
   - 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 29.

After you exit the Application Monitoring Configuration wizard, click the right arrow at the bottom right corner on the Disaster Recovery 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 Cloud Application Host Mapping page, specify the following:

   - Select the on-premises application that you want to map to an Amazon EC2 instance for DR.

     Applications must satisfy certain criteria so that they can be configured for monitoring or DR. Only those applications on the selected on-premises system that satisfy these criteria appear in the **On-premises applications** drop-down list.

     For more information about these criteria, see the *Symantec Disaster Recovery Orchestrator Administration Guide*.

   - Select the name of the Amazon EC2 instance that hosts the application.

     Only those instances that belong to the same AWS availability zone as the Console host appear in the **AWS EC2 instances** table.

   - Enter the user name and password of a domain user who has the privileges to configure the application for DR on the selected instance.

     You may specify the current user or a different user. However, the user must have local administrator privileges on the selected instance.

   Click the Next arrow.

   The wizard searches for application monitoring configurations on the specified instance, 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 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.

For information about the tasks involved in this operation:

The **Retry** link appears next to the failed task. After you resolve the issue, click **Retry** to attempt the task again.

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 25.
- 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).

For information about the tasks involved in this operation:

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

- On the Finalize Application Recovery Configuration panel of the Disaster Recovery 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.
  - 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.
  If a task fails, the **Retry** link appears next to it. Identify and resolve the issue, and then click **Retry** to attempt the task again.

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

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 fully-qualified domain name (FQDN) 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:

- The application may be hosted on a virtual machine at your on-premises site. If the Hyper-V virtualization environment is used, and if live migration is configured, the virtual machine must be configured to use static MAC addresses.

For information about configuring a virtual machine to use static MAC addresses, refer to the Microsoft documentation.

- SQL Server must be installed in the standalone mode in a non-clustered environment. To do so, on the Installation panel of the SQL Server installer, select the **New SQL Server stand-alone installation or add features to an existing installation** option.
- The SQL Server instances must be installed on the local disk. On the Instance Configuration panel of the SQL Server installer, ensure that the **Instance root directory** resides on the local disk.
- If multiple instances of SQL Server exist, each instance must have a unique name instance ID.
- The SQL Server components (FILESTREAM, SQL Server Agent, Analysis Service) that you want to monitor on the system must be installed.
- SQL Server services, apart from the SQL Browser service, must not be set to start at the end of the SQL Server installation. While installing SQL Server on a system, set the startup type of all the SQL Server services to Manual. However, set the startup type of the SQL Server Browser service to Automatic.
- The SQL Server instances that you want to configure for monitoring must not be installed on the system volume, and must be running.
- You can configure application monitoring on a system using the 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.
- Make sure that the user who launches the Application Monitoring Configuration wizard has permissions for the serveradmin and sysadmin roles on SQL Server 2012.

  For more information about the SQL Server roles, see the Microsoft documentation.
- Ensure that the firewall settings at the on-premises site and the security groups in the Amazon VPC allow access to the ports used by the Disaster Recovery Orchestrator installers, wizards, and services.

  For information about the ports used, refer to the *Symantec Disaster Recovery Orchestrator Deployment Guide*.

- After configuring SQL Server for monitoring, if you create another database or service, 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.

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 Configuration wizard. When the Disaster Recovery Configuration wizard does not find an application monitoring configuration on the selected system, it displays a message box accordingly. Click **Configure** to launch the 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 Application Monitoring Configuration wizard.

Note: You can configure monitoring for multiple SQL Server instances 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.

The following panels appear if you select a SQL Server application.

3 On the SQL Server User Details page, provide the credentials of the appropriate SQL Server administrator, and click Next.

This page appears only when you select SQL Server 2012. These credentials are required to discover the databases and services.

Note: The user that you specify on this panel must be a SQL Server administrator for all the instances that you want to configure for monitoring. Only those instances for which that this user is defined as the administrator are discovered, the other instances are not discovered.

4 On the SQL Instance Selection page, specify the following:

- All the instances are selected by default. Deselect only those that you do not wish to monitor.
- All the enabled services are selected by default. Deselect only those that you do not wish to monitor.
- If you wish to perform detail monitoring, select **Configure detail monitoring**, and provide the required input.

Click Next.

5 To enable detail monitoring for the selected instances and associated services, select **Configure detail monitoring** and provide the following required details:

- Enter a non-zero value in the **Monitor after every... cycles** box. This value indicates the number of online monitor cycles that the agent must wait before performing detail monitoring.
  Symantec recommends that you set this value between 1 and 12. The default value is 5.
- Select one of the following modes for detail monitoring:
  - Database monitoring

    Select this mode to enable detail monitoring by choosing the desired databases from a list.

  - Script-based monitoring
    Select this mode to enable detail monitoring by using a user-defined SQL script.
    Enter the location of the SQL script on the system.

- Select **Restart the SQL instance if detail monitoring fails** if you want to detect an application failure in case detail monitoring has failed.

6. 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.

7. 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.

   For further information, refer to the *Symantec Disaster Recovery Orchestrator Administration Guide*.

Appendix A

Troubleshooting

This appendix includes the following topics:

- Application configuration issues
- Configuration of a SQL Server application for recovery fails with sqlcmd errors
- Unable to connect to a SQL Server instance using the system name
- When detail monitoring is configured for SQL Server, the application may fail to come online during takeover or failback if the DNS refresh takes longer

Application configuration issues

The following sections describe some of the issues that you might encounter with the application monitoring and recovery configurations, and provide solutions to work around those issues.

Configuration of a SQL Server application for recovery fails with sqlcmd errors

You might encounter errors related to the `sqlcmd` utility on the Implementation panel of the Disaster Recovery Configuration wizard. The wizard may fail to take the SQL Server application offline on the cloud application host.

Workaround

Check whether the SQL Server client tools are installed on the on-premises application host and the cloud application host.

If the client tools are not installed, check whether the following required patches are installed:

- http://www.microsoft.com/en-us/download/confirmation.aspx?id=36434

- http://www.microsoft.com/en-us/download/confirmation.aspx?id=36433

If these patches are not installed, install them in the order that was mentioned previously. Then, launch the Disaster Recovery Configuration wizard again.

Unable to connect to a SQL Server instance using the system name

By default, the SQL Server instances are identified as follows:

- Default instance: ***SystemName***
  For example: **Sys_01**
- Named instance: ***SystemName\InstanceName***
  For example: **Sys_01\Inst_01**

When configuring SQL Server for recovery in the cloud, the Disaster Recovery Configuration wizard changes its name to the virtual name that you provide on the Virtual Computer Name panel.

This change is not reversed when removing the application recovery configuration or when uninstalling the product. Therefore, you may not be able to access the SQL Server instances using the system name, which is used by default.

Workaround

After removing a recovery configuration from Disaster Recovery Orchestrator Console or after uninstalling Disaster Recovery Orchestrator Client, change the SQL Server name from the virtual name back to the system name.

Perform the following steps on the on-premises application host and the corresponding cloud application host.

To restore the default SQL Server name

1. Launch the Query Analyzer.
   - On Windows Server 2008 R2, click **Start > All Programs > Microsoft SQL Server > Query Analyzer**.
   - On Windows Server 2012, open Query Analyzer from the **Apps** menu.
2. In the Connect to SQL Server window, provide the following information:
   - In the **SQL Server** field, enter the SQL Server machine name in the format ***VirtualName\InstanceName***. For example, **DR_DB\Inst_01**.
   - Select the **Start SQL server if it is stopped** checkbox.

- Select either **Windows authentication** or **SQL Server authentication**, and enter the valid values for **Login name** and **Password**.

Click **OK**.

3 In the SQL Query Analyzer window, find the SQL Server name as follows:

- In the Editor pane, enter:

  ```
  sp_helpserver
  ```

- Press F5.
- Make note of the name listed in the Results pane, for example, DR_DB\Inst_01.
  For a named instance, the name will be *VirtualName\InstanceName*. For a default instance, the name will be *VirtualName*.

4 Delete the contents of the Editor pane.

5 Remove the current SQL Server name as follows:

- In the Editor pane, enter:

  ```
  sp_dropserver ‘VirtualName\InstanceName'
  ```

  Here, *VirtualName\InstanceName* should be the value that noted in step 3.
  For example, for a named instance enter:

  ```
  sp_dropserver ‘DR_DB\Inst_01’
  ```

  For example, for a default instance enter:

  ```
  sp_dropserver ‘DR_DB’
  ```

- Press F5.

6 Delete the contents of the Editor pane.

7 Add the new SQL Server name as follows:

- In the Editor pane, enter:

  ```
  sp_addserver 'SystemName\InstanceName', local
  ```

  For example, for a named instance enter:

  ```
  sp_addserver ‘Sys_01\Inst_01’, local
  ```

  For example, for a default instance enter:

  ```
  sp_addserver ‘Sys_01’, local
  ```

- Press F5.

When detail monitoring is configured for SQL Server, the application may fail to come online during takeover or failback if the DNS refresh takes longer

The last task of the takeover or failback operation is to bring the application online on its host. To do so when detail monitoring is configured, Disaster Recovery Orchestrator attempts to connect to the SQL Server instance using its virtual name. The application host may not receive the updated DNS record for the virtual name in time. If this happens, the connection to the SQL Server instance fails. You might encounter errors such as the following in the agent log, `SQLServer_A.txt`:

```
FaultOnDMFailure was set and DetailMonitor of DBList failed.
Returning OFFLINE...
```

```
Failed to open a ADO Connection to the DataBase 'DBInstanceName'.
```

```
SQL Server does not exist or access denied.
```

Workaround

If you encounter these errors, perform the following tasks:

1. Log on to the system where the application needs to be brought online and test the connection to the virtual name by running the following command:

   ```
   ping AppVirtualName
   ```

   Replace the *AppVirtualName* variable with the value that you provided on the Virtual Computer Name panel when configuring the application for recovery.

2. Depending on the output, do one of the following:

   - If you receive a message similar to the following, wait for a few moments and then run the command again

     ```
     Ping request could not find host AppVirtualName.
     Please check the name and try again
     ```

   - If you receive a response to the ping command, the connection to the application using its virtual name is successful.
     On the Console UI, click **Retry** against the failed task; if there are no other issues, the application comes online.

Index