

SQL Server Replication for Cisco Unity Failover

Document ID: 91961

Introduction

Prerequisites

- Requirements
- Components Used
- Conventions

Monitor Replication

- Causes of SQL Replication Failure
- Use the Event Log to Monitor the SQLSERVERAGENT Service
- Use the Event Log to Monitor Replication Jobs
- Use SQL Server Enterprise Manager to Monitor SQLSERVERAGENT Replication Jobs
- Determine Whether Pending Replication Transactions Are Being Processed

Restart Replication Jobs

- Start Replication Jobs that Failed to Start

Disable and Re-enable Replication

- Disable Automatic Failover, and Stop File and SQL Replication
- Configure Failover on the Primary Server
- Configure Failover on the Secondary Server

Change which Accounts Own Replication Jobs

- Change the Account that Owns Replication Jobs

Further Improvements for Replication Monitoring

NetPro Discussion Forums – Featured Conversations

Related Information

Introduction

When failover is configured, Cisco Unity uses Microsoft SQL Server 2000 replication to replicate data from the active server to the inactive server. If failover occurs, data replication ensures that current configuration and subscriber data is available on the secondary server and that, after failback, changes made on the secondary while it was active are replicated back to the primary. Replication is performed by SQL Server replication jobs, which are run by the SQLSERVERAGENT service.

When SQL Server replication breaks, replication transactions are saved in audit log tables on the active server so the data can be replicated to the inactive server when replication is restored. If replication is broken for an extended period, the audit log tables can become large. This can cause performance degradation, which in turn can cause poor TUI response and can even cause failover to occur. Moreover, the Cisco Unity database on the inactive server does not have the latest configuration and subscriber data when it becomes the active server.

Prerequisites

Requirements

Make sure that you have completed the Requirements for Cisco Unity Failover before you configure the Cisco Unity Failover.

Components Used

The information in this document is based on Cisco Unity 4.0(3) through 4.2(1).

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Conventions

Refer to Cisco Technical Tips Conventions for more information on document conventions.

Monitor Replication

Causes of SQL Replication Failure

Replication can fail if the SQLSERVERAGENT service fails to start and/or if the replication jobs that the service runs fail to start. These failures can occur when SQL Server restarts (for example, when the Cisco Unity server is rebooted) as a result of timing issues, patches, or the application of security or computer policies.

Use the Event Log to Monitor the SQLSERVERAGENT Service

If the SQLSERVERAGENT service fails to start, an event is logged in the system event log. For example:

```
Event Type:      Error
Event Source:    Service Control Manager
Event Category:  None
Event ID:        7001
Date:           5/4/2007
Time:           10:58:47 PM
User:           N/A
Computer:       <servername>
Description:
The SQLSERVERAGENT service depends on the MSSQLSERVER service
which failed to start because of the following error:
The account name is invalid or does not exist, or the password
is invalid for the account name specified.
```

In addition, Cisco Unity detects the problem and logs this event in the application event log. It is recommended that you create email or pager notifications based on this event using any event-monitoring service. For example, the Cisco Unity Event Monitoring Service.

```
Event Type:      Warning
Event Source:    CiscoUnity_NodeMgr
Event Category:  Run
Event ID:        1006
Date:           1/1/2007
Time:           9:00:00 AM
User:           N/A
Computer:       <servername>
Description:
The SQL Server Agent service is not running. It must be running
in order for replication to take place.
```

Use the Event Log to Monitor Replication Jobs

When the jobs that the SQLSERVERAGENT service runs fail to start, by default, no event is logged in the event log. Cisco recommends that you:

- Complete the steps in the Configure SQL Server to Log an Event When a Replication Job Fails to Start section of this document in order to configure SQL Server to log an event in the application event log when replication jobs fail to start.
- Create email or pager notifications based on these events using any event-monitoring service. For example, the Cisco Unity Event Monitoring Service.

Configure SQL Server to Log an Event When a Replication Job Fails to Start

Complete these steps:

1. Start SQL Server Enterprise Manager.
2. In the left pane of Enterprise Manager, expand **Microsoft SQL Servers > SQL Server Group > (local) (Windows NT) > Management > SQL Server Agent**, and click **Jobs**.
3. In the right pane, right-click the name of the job and choose **Properties > Notifications**.

See Table 1 for a list of the replication jobs started by the SQLSERVERAGENT service.

4. In the <job> Properties dialog box, go to the Notifications tab.
5. Check the **Write to Windows Application Event Log** check box.
6. Click **OK** to close the <job> Properties dialog box.
7. Repeat steps 3 through 6 for the rest of the jobs for which you want to log events.

Table 1 Replication Jobs Started by the SQLSERVERAGENT Service

| Replication Job | Description | Category |
|---|---|---------------------------|
| Reinitialize subscriptions having data validation failures. | Reinitializes all subscriptions that have data validation failures. | REPL-Alert Response |
| Replication agents checkup | Detects replication agents that do not log history actively. | REPL-Checkup |
| SVRNAME-UnityDb-UnityDbPublication-SVRNAME-3 | UnityDb Distribution | REPL-Distribution |
| Distribution clean up: UnityDistributionDb | Removes replicated transactions from the distribution database. | REPL-Distribution Cleanup |
| Agent history clean up: UnityDistributionDb | Removes replication agent history | REPL-History Cleanup |

| | | |
|--------------------------------------|---|---------------------------|
| | from the distribution database. | |
| SVRNAME–UnityDb–3 | UnityDb LogReader | REPL–LogReader |
| [SVRNAME].9 | Reads queues for Queued updating subscriptions. | REPL–QueueReader |
| SVRNAME–UnityDb–UnityDbPublication–3 | UnityDb Snapshot | REPL–Snapshot |
| Expired subscription clean up | Detects and removes expired subscriptions from published databases. | REPL–Subscription Cleanup |

Use SQL Server Enterprise Manager to Monitor SQLSERVERAGENT Replication Jobs

Use Enterprise Manager to Determine Whether Replication Jobs Succeed

Complete these steps:

1. On the primary server, start SQL Server Enterprise Manager.
2. In the left pane of Enterprise Manager, expand **Microsoft SQL Servers > SQL Server Group > (local) (Windows NT) > Management > SQL Server Agent** and click **Jobs**.
3. In the right pane, each job has an icon that indicates its success or failure. Any job with a red-dot icon has failed. For any jobs that fail, if the value of the Status column is:
 - ◆ Executing The red-dot icon has not been updated with the final status. Wait until the icon has been updated.
 - ◆ For any other value, right-click on the job name, and click **View Job History** in order to display the reason that the job failed.

Determine Whether Pending Replication Transactions Are Being Processed

During a replication outage, replication transactions can accumulate until there are more than can be processed while the system handles a normal call volume. (The most common example of this is an ODBC timeout when the primary and secondary Cisco Unity servers attempt to connect to one another.) After the outage, when you allow the replication jobs to run during a relatively slow time (such as over night or over a weekend), the replication jobs can often clear the backlog of unreplicated transactions. However, if there are a lot of unreplicated transactions, attempts by SQL Server to replicate the data can result in a timeout. If replication is functioning but the number of unreplicated transactions has not dropped significantly by the end of a weekend, you might need to disable and then re-enable replication. See the Disable and Re-enable Replication section of this document for more information.

Use the OSQL commands in this section to determine whether the number of unreplicated transactions is unusually large after an outage and whether the oldest transactions are being processed. (For a system with a large number of Cisco Unity subscribers and a lot of activity, transactions that range in the hundreds can be common. Transactions that range in the thousands are cause for concern.)



Caution: If the number of unreplicated transactions is very large, the OSQL commands might take a long time to complete and put considerable additional load on the server.

Complete these steps in order to display an ordered list of the dates on the pending replication records, which you can use to determine how old the oldest transaction is:

1. On the secondary server, choose **Start > Run**.
2. Run **cmd**.
3. At the command prompt, run this command in order to start OSQL and query the Unitydb database:

Note: This command is wrapped to a second line due to spatial reasons.

```
OSQL -E -d Unitydb -Q "SELECT distinct insertdate
FROM MSreplication_queue ORDER BY insertdate"
```

Note: OSQL switches are case sensitive (for example, -E).

Complete these steps in order to obtain a total count of pending replication records. You can run these records daily to determine whether the number of unreplicated transactions is growing or shrinking:

1. On the secondary server, choose **Start > Run**.
2. Run **cmd**.
3. At the command prompt, run this command in order to start OSQL and query the Unitydb database:

```
OSQL -E -d Unitydb -Q "SELECT count(*) FROM MSreplication_queue"
```

Note: OSQL switches are case sensitive (for example, -E).

Complete these steps in order to determine whether data from the primary server is replicated to the secondary server:

1. On the primary server, choose **Start > Run**.
2. Run **cmd**.
3. At the command prompt, run this command in order to start OSQL and query the UnityDistributionDb database:

Note: This command is wrapped to a second line due to spatial reasons.

```
OSQL -E -d UnityDistributionDb -Q "SELECT SUM(UndelivCmdsInDistDB)
FROM MSdistribution_status"
```

Restart Replication Jobs

Usually, if the SQLSERVERAGENT service or one of the replication jobs fails to start, it is due to a timing issue during startup. You can generally restore replication when you start any jobs that failed to start.

Start Replication Jobs that Failed to Start

Complete these steps:

1. Start SQL Server Enterprise Manager.
2. In the left pane of Enterprise Manager, expand **Microsoft SQL Servers > SQL Server Group > (local) (Windows NT) > Management > SQL Server Agent** and click **Jobs**.
3. Right-click the job that failed to start and click **Start Job**.
4. If the value of the Status column does not change to Executing, review the job history. Right-click the job, and click **View Job History**. When the cause of the failure is corrected, repeat step 3 in order to start the job.

Disable and Re-enable Replication

If the number of unreplicated transactions is so large that the replication jobs repeatedly time out, and if this prevents replication from significantly reducing the number of unreplicated records, you must disable and then re-enable replication. Complete these three procedures in order to accomplish this:

1. Disable Automatic Failover, and Stop File and SQL Replication
2. Configure Failover on the Primary Server
3. Configure Failover on the Secondary Server



Caution: If you disable and re-enable replication, the unreplicated transactions (if any) on both the primary and secondary servers are deleted, and the Cisco Unity database on the primary server is replicated to the secondary server. If there are any unreplicated changes on the secondary server, those changes are lost.

Disable Automatic Failover, and Stop File and SQL Replication

Complete these steps:

1. If the primary server is active, proceed to step 5.

If the primary server is not active, on the secondary server choose **Start > Programs > Cisco Unity > Failover Monitor**.

2. Click **Failback**.
3. Click **OK** to confirm that you want to fail back to the primary server.
4. Close the Failover Monitor.
5. On the primary server, on the Windows Start menu, choose **Programs > Cisco Unity > Failover Monitor**.
6. Click **Advanced**.
7. Check the **Disable Automatic Failover and Failback** check box.
8. Click **OK** and close the Failover Monitor.
9. On the primary server, on the Windows Start menu, choose **Programs > Administrative Tools > Services**.
10. In the right pane, double-click **AvCsNodeMgr**.
11. On the General tab, click **Stop**.
12. In the Startup Type list, click **Disabled**.
13. Click **OK**.
14. Close the Services window.



Caution: Because the Node Manager service is disabled, file replication stops. Replication is

- re-enabled when normal failover operation resumes.
15. On the secondary server, on the Windows Start menu, choose **Programs > Administrative Tools > Services**.
 16. In the right pane, double-click **AvCsNodeMgr**.
 17. On the General tab, click **Stop**.
 18. In the Startup Type list, click **Disabled**.
 19. Click **OK**.
 20. Close the Services window.
 21. On the primary server, on the Windows Start menu, choose **Programs > Microsoft SQL Server > Enterprise Manager**.
 22. In the left pane of the Console Root window, browse to the Replication node for the primary server. Typically, the node is three levels under the Microsoft SQL Servers node.
 23. Right-click the **Replication** node, and click **Disable Publishing**. The Disable Publishing and Distribution wizard appears.
 24. On the Welcome page, click **Next**.
 25. On the Disable Publishing page, click **Yes**, then click **Next**.
 26. On the Confirm Dropping of Publications page, click **Next**.
 27. On the Completing page, click **Finish**.
 28. When the process is complete, click **OK**.
 29. Close the Console Root window.
 30. Exit Enterprise Manager.

Configure Failover on the Primary Server

Complete these steps:

1. In Windows Explorer, browse to the CommServer directory.
2. Double-click **FailoverConfig.exe** to start the Configure Cisco Unity Failover wizard.
3. On the Welcome page, click **Next**.
4. On the Specify Server Role page, click **Primary Server**, and click **Next**.
5. On the Enter the Name of Your Server page, click **Browse**, select the name of the secondary server, and click **OK**. The IP address for the secondary server is filled in automatically.
6. Click **Next**.
7. On the Enter Failover Account Information page, click **Browse**, and double-click the name of the message store services account. This is the account that the failover service logs on as.

The account you select must have the right to act as part of the operating system and to log on as a service, and must be a member of the Local Administrators group.



Caution: You must specify the same account on both the primary and secondary servers.

8. In the Password field, enter the password for the account that the failover service logs on as, and click **Next**.
9. On the Begin Configuring Your Server page, click **Configure**. The wizard verifies settings and configures failover on the primary server.

If the wizard does not finish the configuration successfully, an error message explains why the wizard failed. Exit the wizard, correct the problem, and click **Configure** again.

10. On the Completing page, click **Finish**.

Configure Failover on the Secondary Server

Complete these steps:

1. On the Windows taskbar, double-click the system clock. The Date/Time Properties dialog box appears.
2. Set the time to the same hour and minute as shown on the primary server, and click **OK**.
3. In Windows Explorer, browse to the **CommServer** directory.
4. Double-click **FailoverConfig.exe** to start the Configure Cisco Unity Failover wizard.
5. On the Welcome page, click **Next**.
6. On the Specify Server Role page, click **Secondary Server**, and click **Next**.
7. On the Enter the Name of Your Server page, click **Browse**, select the name of the primary server, and click **OK**. The IP address for the primary server is filled in automatically.
8. Click **Next**.
9. On the Enter Failover Account Information page, click **Browse**, and double-click the name of the message store services account. This is the account that the failover service logs on as.

The account you select must have the right to act as part of the operating system and to log on as a service, and must be a member of the Local Administrators group.



Caution: You must specify the same account on both the primary and secondary servers.

10. In the Password field, enter the password for the account that the failover service logs on as and click **Next**.
11. On the Begin Configuring Your Server page, click **Configure**. The wizard verifies settings and configures failover on the secondary server.

If the wizard does not finish the configuration successfully, an error message explains why the wizard failed. Exit the wizard, correct the problem, and click **Configure** again.

12. On the Completing page, click **Finish**.

Change which Accounts Own Replication Jobs

By default, Windows domain accounts own replication jobs. This adds some complexity by introducing dependencies on Windows authentication and on networking communication. SQL Server has two built-in accounts that are not Windows domain accounts and are unique to SQL Server. In order to reduce dependencies, change ownership of the replication jobs to one of these SQL Server accounts:

- The sa account is the built-in SQL Server administrative account. This account has a high level of access.
- The distributor_admin account is created when replication is configured. This account has a lower level of access than the sa account.

Change the Account that Owns Replication Jobs

Complete these steps:

1. Start SQL Server Enterprise Manager.
2. In the left pane of Enterprise Manager, expand **Microsoft SQL Servers > SQL Server Group > (local) (Windows NT) > Management > SQL Server Agent**, and click **Jobs**.
3. For the first replication job listed in Table 1, right-click on the job, and click **Properties**.
4. On the General tab, in the Owner list, click the name of the account that you want to own the job. Cisco recommends that you choose the distributor_admin account.
5. Click **OK** to close the <job> Properties dialog box.
6. Repeat steps 3 through 5 for the rest of the jobs in Table 1.
7. Restart all of the replication jobs:

- a. For the first replication job listed in Table 1, right-click the job and click **Stop Job**.
- b. Right-click the job and click **Start Job**.
- c. Repeat steps a and b for the rest of the jobs in Table 1.

Further Improvements for Replication Monitoring

One outstanding issue with monitoring SQL Server replication jobs is that some jobs only start once, when SQL Server and the SQLSERVERAGENT service are started. As a result, if the jobs fail, they only cause one event to be logged. (Other replication jobs start, stop to "go to sleep," and then restart. These jobs log an error each time they fail to start.)

In order to continually monitor the status of jobs that only start once, the Cisco Unity engineering group adds monitoring of replication jobs to the existing monitoring of the SQLSERVERAGENT service, as described earlier in this document. This improvement is tracked with Cisco bug ID CSCsi50517 (registered customers only)

NetPro Discussion Forums – Featured Conversations

Networking Professionals Connection is a forum for networking professionals to share questions, suggestions, and information about networking solutions, products, and technologies. The featured links are some of the most recent conversations available in this technology.

| |
|---|
| NetPro Discussion Forums – Featured Conversations for Voice |
| Service Providers: Voice over IP |
| Voice & Video: Voice over IP |
| Voice & Video: IP Telephony |
| Voice & Video: IP Phone Services for End Users |
| Voice & Video: Unified Communications |
| Voice & Video: IP Phone Services for Developers |
| Voice & Video: General |

Related Information

- [Voice Technology Support](#)
- [Voice and Unified Communications Product Support](#)
- [Recommended Reading: Troubleshooting Cisco IP Telephony](#)
- [Technical Support & Documentation – Cisco Systems](#)

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2007 – 2008 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

Updated: Jun 12, 2007

Document ID: 91961