Abstract
This white paper discusses how EMC® Replication Manager integrates with Microsoft SQL Server to provide a solution for continuous availability of critical user data.
March 2012
# Table of Contents

Executive summary ............................................................................................................ 5  
Introduction ....................................................................................................................... 5  
  Audience ......................................................................................................................... 6  
Replication Manager architecture ....................................................................................... 6  
  Replication Manager Console and CLI ............................................................................ 7  
  Replication Manager Server ......................................................................................... 7  
  Replication Manager agents ......................................................................................... 8  
  Replication Manager SQL Server agent ........................................................................ 8  
Storage technologies ......................................................................................................... 8  
  CLARiiON replication ...................................................................................................... 8  
  Symmetrix replication ..................................................................................................... 9  
  Celerra replication ......................................................................................................... 9  
  RecoverPoint family ...................................................................................................... 10  
System requirements ......................................................................................................... 10  
Security considerations ..................................................................................................... 10  
  Preparing the SQL Server production server ................................................................. 10  
  Optimal database configuration ..................................................................................... 11  
Preparing the SQL Server backup (mount) server ............................................................ 12  
SQL Server recovery models ............................................................................................. 12  
Creating SQL Server replicas ........................................................................................... 13  
  Truncating the transaction log ....................................................................................... 13  
  Details of creating a SQL Server replica ....................................................................... 14  
  Creation of replicas using the VDI snapshot API ......................................................... 15  
Mounting SQL Server replicas ......................................................................................... 16  
  Mounting for backup .................................................................................................... 16  
  Mounting for reuse ....................................................................................................... 17  
Restoring SQL Server databases ....................................................................................... 17  
  Restoring from the replica ........................................................................................... 17  
  Restoring one database to a volume with many databases .......................................... 18  
    How to run rmsqlrestore ............................................................................................ 18  
    Steps for restoring with rmsqlrestore ....................................................................... 18  
    Preforming the file or filegroup restore ................................................................... 19  
    The rmsqlrestore command syntax ........................................................................ 20  
    Examples .................................................................................................................... 20  
Special Considerations ..................................................................................................... 21
Conclusion ................................................................................................................................. 21
References ............................................................................................................................... 21
Executive summary

EMC® Replication Manager is a software application that simplifies the creation and management of disk-based replicas. Replication Manager automates the creation of exact point-in-time copies of mission-critical data that can be used for rapid recovery or repurposing.

Replication Manager includes specific benefits for improving availability and data protection:

- **Fast backup and restore** - Using EMC storage technologies, Replication Manager can back up and restore databases almost instantly.

- **Automated replication** - Replication Manager allows DBAs to schedule application consistent SQL Server replications to occur at regular intervals and decide the lifecycle of those replicas.

- **Mounting replicas** - A database can automatically be mounted to alternate hosts based on replication policies set within the job.

- **Mounting for backup or repurposing** - Replication Manager can automatically mount a SQL Server database to an alternate server and run a backup program to make a copy of the files for long-term archival, offloading the task from the production host. Or the database can be mounted and used for report generation or data mining.

This white paper focuses on Replication Manager's application integration with Microsoft SQL Server 2000 and SQL Server 2005.

Introduction

You can use Replication Manager to mount a replica to another server and recover the database. The database can then be used for other activities like testing or reporting. In addition, Replication Manager is a backup and restore accelerator. Replication Manager can quickly create clones or snapshots of your production database with virtually no impact on performance. It can also be used to quickly restore a database.

There are many solutions to the issue of database corruption. The traditional approach is to restore from tape backups. This solution may be acceptable in some cases, but for business systems that require speedy and reliable data restoration, tape is not the most effective answer, due to several deficiencies:

- **Frequency of tape backup** — Differential backups to tape are typically taken once a day and full backups to tape are taken once a week. Transaction log backups are used to fill in the gaps during the day. Recovery of a database requires the full backup, the most recent differential backup, and all of the transaction log backups taken after the most recent differential - it requires the coordination of restores from several different sources.
• **Location of tape** — Tape archives are generally kept offsite in a safe environment. This means that the tapes for the last full backup and any differential backups must be identified and retrieved from the remote site, increasing the delay in data restoration.

• **Time to restore** — The tapes must be applied sequentially to restore the data. Again, this is a time-consuming task.

Companies competing in e-commerce, share trading, and other time-critical business sectors cannot afford lengthy data outages while data centers restore tape backups. In the event of a media or logical corruption, they need quick access to reliable, up-to-date data sets.

Today’s information technology professionals must provide alternate solutions that offer quick, reliable data replication and recovery. This white paper discusses how Replication Manager can offer such a solution by providing continuous availability to critical business data.

**Audience**

This white paper has been written for the following audiences:

• Microsoft SQL Server administrators responsible for implementing Replication Manager in a SQL Server environment

• EMC internal and field personnel who assist customers with implementing Replication Manager in Microsoft SQL Server environments

**Replication Manager architecture**

Replication Manager uses a LAN and SAN to communicate and control storage-based functions. Figure 1 shows the Replication Manager architecture and the components that reside in various parts of the system. An introduction to each component is provided.
Figure 1. Replication Manager software architecture

**Replication Manager Console and CLI**

The Replication Manager Console is a portable Java application that lets you control Replication Manager from a Windows system that has a TCP/IP connection to the Replication Manager Server.

A command line interface is also provided. It can be run interactively or in batch mode.

**Replication Manager Server**

The server software is installed on a Windows system. It controls replication activities and stores data about each replica. The server software has three distinct components:

- **Replication Manager Server Service**, also known as the IRD, controls and coordinates replication and recovery activity for all storage corresponding to registered clients and their application sets. The IRD also handles all requests from the Replication Manager Console.

- **Policy Engine** links Replication Manager with the supported storage technologies. The Replication Manager Server Service uses the policy engine to analyze the storage environment and select the appropriate storage for the replica.

- **Replication Manager Repository** is an embedded database that stores data about application sets (for instance, information about SQL Server databases and filegroups), jobs (backup options, mount options, and so on), and replicas.

When installed on Microsoft Cluster Server, Replication Manager Server is a cluster-aware application and its components will fail over to a passive node in the event of a cluster failover.
Replication Manager agents

Replication Manager agent software is installed on each host that participates in the replication process, including hosts that manage production data and hosts that are used to mount and back up replicas. The agent software has three distinct components:

- **Replication Manager Client Service**, also known as IRCCD, waits for incoming requests from the IRD, then coordinates all operations on the agent.
- **Storage services** component manages the storage relationships between the client service and the storage technologies used to create the replicas.
- **Application agents** are provided for each supported file system and database application, including Microsoft SQL Server. Each agent is a separate dynamic library (DLL) that is loaded by the client service as needed. The following section describes the SQL Server agent in detail.

Replication Manager SQL Server agent

To support SQL Server, Replication Manager uses Microsoft SQL Server’s Virtual Device Interface (VDI) snapshot API. As with agents for other applications, the SQL Server agent provides Replication Manager with a logical view of the data that resides on the storage array. The agent enables Replication Manager to:

- Specify which databases or filegroups to replicate.
- Ensure that the data can be replicated safely and quickly.
- Return the database to normal operation after creating the replica.
- Recover a database on another host so it can be used for other operations such as testing, reporting, or data mining.
- Quickly recover a database on the production host in the event of data corruption.

Storage technologies

This section summarizes the storage technologies supported by Replication Manager and how they relate to SQL Server replication.

CLARiiON replication

On CLARiiON® arrays, Replication Manager can take advantage of both clone and snapshot functionality. Clones are fractured from the source and therefore have no impact on production environments, and are ideal for creating copies of production data for business continuity, backup acceleration, and repurposing. Snapshots offer an alternative to clones in that they require less space, but because they are pointer-based they may cause spindle contention, in which case users may opt for the performance benefit provided by clones. Since both features are included with SnapView™, customers can evaluate which makes the most sense for their particular
environment. CLARiiON-based storage technologies to consider for SQL Server replication are:

- CLARiiON SnapView clone, which creates exact copies of active production data on separate drives.
- CLARiiON SnapView clone (remote), which creates clone LUNs of a MirrorView/S secondary device (Targets that use multiple secondaries are not supported by Replication Manager.)
- CLARiiON SnapView snapshot, which creates SnapView snapshots of CLARiiON LUNs
- CLARiiON SnapView snapshot (remote), which creates SnapView snapshots of a MirrorView/S secondary device (Targets that use multiple secondaries are not supported by Replication Manager.)
- SAN Copy™, which is a storage system application that copies data between storage arrays (Symmetrix®-to-CLARiiON and CLARiiON-to-CLARiiON) and within a CLARiiON array

**Symmetrix replication**

On Symmetrix arrays, Replication Manager can take advantage of TimeFinder®/Mirror (BCVs), TimeFinder/Clone (STDs), or TimeFinder/Snap (VDEVs). Symmetrix-based storage technologies to consider for SQL Server replication are:

- TimeFinder/Mirror, which creates mirror images of active production volumes from a standard device onto a BCV device
- TimeFinder/Mirror (remote), which creates mirror images of active production volumes from a standard device onto BCVs for R2 devices across an SRDF link
- TimeFinder/Clone, which creates copies of active production volumes in which the source and target devices can be either STD or BCV devices
- TimeFinder/Clone (remote), which creates copies across an SRDF link from an R1 device onto a STD or BCV on a remote Symmetrix
- TimeFinder/Snap, which creates copies of active production volumes or R1 devices onto a TimeFinder/Snap (also known as a VDEV) on the local Symmetrix
- Symmetrix TimeFinder/Snap (remote), which creates copies of active production volumes onto TimeFinder/Snaps (VDEVs) by way of an R2 device located on a remote Symmetrix array (connected to the production volumes using SRDF/S)
- SAN Copy (Symmetrix-to-CLARiiON)

**Celerra replication**

On Celerra® iSCSI network servers, Replication Manager can take advantage of Celerra SnapSure™ for local snapshots or Celerra Replicator™ for remote snapshot functionality. When you create a replica using iSCSI Celerra snapshot functionality,
Replication Manager creates a point-in-time copy of all data on the source LUN. For the initial snapshot, Replication Manager creates a full copy of the original LUN, therefore requiring the same amount of space on the file system as the LUN. The amount of space used by subsequent snapshots depends on how much the data has changed since the last snapshot was taken. For copy replicas, a full copy is created each time the job runs.

**RecoverPoint family**

Replication Manager supports EMC RecoverPoint software, enabling users to manage applications protected by RecoverPoint Continuous Data Protection (CDP), RecoverPoint Continuous Remote Replication (CRR), and RecoverPoint Concurrent Local Remote Replication (CLR). Replication Manager’s support for RecoverPoint provides near-instantaneous mounting of critical business data. It allows rapid recovery of the data on a volume to any point in time (crash-consistent recovery) or to a significant point in time (application-consistent recovery). Replication Manager with RecoverPoint addresses the problem of lengthy recoveries of files and databases made by traditional backup software. It allows rapid reconstruction of a volume image that may be only seconds or minutes old.

**System requirements**

Replication Manager supports SQL Server 2000, SQL Server 2005, and SQL Server 2008. For the latest information on system requirements, refer to the E-Lab™ Interoperability Navigator on Powerlink®.

**Security considerations**

Replication Manager and SQL Server require certain permissions and rights to configure application sets, run jobs, and recover databases. When you create an application set and when you recover a database during a mount, Replication Manager asks for a user account. The account can use either SQL Server authentication or Windows authentication. This account is used to discover databases, create replicas of the databases, and to recover the databases. If you use a Windows user account, Replication Manager impersonates that account when creating or restoring a replica. The account can either be a member of the local Administrators group, or a domain or local user account. To use an account that is not a member of the local Administrators group, a QFE is required from Microsoft; refer to KB934396 for more information. The *EMC Replication Manager Product Guide* describes how to set up this account.

**Preparing the SQL Server production server**

Note the following when preparing the SQL Server production server to operate with Replication Manager:
- All data files (including full text catalogs for SQL Server 2005) and transaction log files must reside on replicable storage.

- System databases and user databases should be on separate volumes. The recovery process for a system database is different from that of a user database.

- Refer to the E-Lab Interoperability Navigator on Powerlink for information about required service packs and hot fixes.

- For performance reasons, the data and log files should be on different volumes (this is not a requirement, however).

- Because Replication Manager creates and restores replicas at the volume level, databases should be on their own volumes. Because this is not always possible, Replication Manager has a command line tool (rmsqlrestore) that can help you restore an individual database rather than the whole volume. Refer to "How to run rmsqlrestore" on page 18.

**Optimal database configuration**

Some large databases are spread across many devices, and are logically divided into filegroups. These filegroups can represent different aspects of a business that contain tables and data specific to that group. For example, the Billing department has its own set of tables that Payroll does not need to access, and vice versa. Replication Manager allows for individual backup and restore of filegroups. Each filegroup is made up of one or more physical files, for example, `i:\SymmStorage\MyProdDB_payroll.mdf`. A file can only belong to one filegroup, but a filegroup can be made up of several physical files.

<table>
<thead>
<tr>
<th>Filegroup</th>
<th>File(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Payroll</td>
<td><code>i:\SymmStorage\MyProdDB_payroll.mdf</code></td>
</tr>
<tr>
<td>Billing</td>
<td><code>k:\SymmStorage\MyProdDB_billing.mdf</code></td>
</tr>
</tbody>
</table>

The active log (.ldf) is common to all filegroups, so on a large database where filegroup-level restore is possible, the log must be kept on its own device. The active log does not belong to any filegroup. It must be backed up periodically to facilitate rolling forward the database after restoring a single or subset of filegroups. Use the SQL Server Management Studio or Enterprise Manager to schedule a maintenance job to back up the transaction logs.

Active Log `m:\SymmStorage\MyProdDB.ldf`

EMC recommends that each filegroup has its own device. The active log should have a separate device. If storage is an issue and two or more filegroups share a device, then all filegroups located on that device will always be restored simultaneously.
Preparing the SQL Server backup (mount) server

Replication Manager can mount a replica to the production server (assuming it is not a cluster) or to another server. If you want to recover a replica on another server to run a consistency check (DBCC) or for repurposing, SQL Server must be installed on the other server. It is recommended that the exact same version of SQL Server be installed on both servers; however, SQL Server will allow you to use a newer version of SQL Server. If you recover a replica with a newer version of SQL Server, you will not be able to recover that replica to the production server.

SQL Server recovery models

SQL Server uses recovery models to simplify database administration. The recovery models and related Replication Manager requirements and limitations are as follows:

- **Simple Recovery model** — SQL Server performs minimal transaction log maintenance in this mode. It protects the database from events such as server restart, but because there is no chain of transaction logs to replay, it cannot recover from a backup and roll forward.

  When the Simple Recovery model is used, Replication Manager supports only full backup and restore; filegroup and transaction log backup and restore are not supported.

- **Bulk-Logged Recovery model** — This recovery model mitigates problems caused by very large transaction logs. Instead of fully logging all operations, SQL Server performs minimum logging for bulk imports.

  With the Bulk-Logged Recovery model, Replication Manager supports full database and full filegroup backups and restores. Replication Manager also supports backing up the transaction logs before doing a restore when using this recovery model.

- **Full Recovery model** — In this model, all updates to the database are recorded in the transaction log. The continuous sequence of transaction logs allows for recovery of a database to nearly any required point.

  With the Full Recovery model, Replication Manager supports full database and full filegroup backups and restores. Replication Manager also supports backing up the transaction logs before doing a restore when using this recovery model.

If you are using the consistent split option with the Online with limited recovery consistency method (refer to Figure 2), you will have limited restore options. Because Replication Manager makes a copy of the database using technology in the array and does not use VDI, you cannot roll the database forward. Once Replication Manager restores the data on the volume, you can attach the database. You cannot:

- Use any of SQL Server’s recovery options
- Restore individual filegroups
- Back up the transaction log before doing a restore.

![Advanced Replication Settings](image)

**Figure 2. Advanced Replication Settings for a non-VDI Consistent Split Job**

**Creating SQL Server replicas**

In general, Replication Manager performs online backup of databases using VDI. First, Replication Manager prepares the target storage for replication. This usually means the production volumes are synchronized with the target storage. VDI is then used to freeze I/O to the database, the target storage is split from the production storage, and VDI is used to thaw the I/O. A VDI metadata file is a product of the snapshot process. It contains information about the database. This is necessary for Replication Manager to recover the data in the replica.

This metadata file is sent to the Replication Manager Server for safekeeping so it can be accessed in the event of a restore or a mount.

Replication Manager can be used to schedule these online backups or a third-party scheduler can be used. The easiest way to use a third-party scheduler is to create a schedule with Replication Manager and then copy the command that is generated from the Windows scheduler to the third-party scheduler.

**Truncating the transaction log**

In addition to scheduling the online backups, you should create a maintenance plan for the transaction logs so they are truncated on a regular basis. If log records were never deleted from the transaction log, the logical log would grow until it filled all the available space on the disks holding the physical log files. At some point, old log
records no longer necessary for recovering or restoring a database must be deleted to make way for new log records. The process of deleting these log records to reduce the size of the logical log is called truncating the log.

You can create a maintenance plan using SQL Server’s management tools or you can schedule BACKUP LOG statements to occur at intervals that will keep the transaction log from growing past the desired size. This means that the sequence of log backups must contain every log record that was written since the database backup. When you are maintaining a sequence of transaction log backups, no log record can be truncated until after it has been written to a log backup. Active logs (.ldf) and transaction log backups must not be confused. The latter is a production of backing up the former. Only the transaction log backup can be used in the roll-forward procedure.

**Details of creating a SQL Server replica**

Replication Manager creates replicas of SQL Server 2000 and 2005 data using either the Microsoft SQL Server VDI snapshot API or consistent split technology in the array. This section focuses on the VDI snapshot support, but touches on the differences between replicas created with VDI and those created without VDI.

**Table 1. Comparison of VDI snapshot and consistent split**

<table>
<thead>
<tr>
<th>VDI snapshot</th>
<th>Consistent split only (with no VDI)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Can back up databases or filegroups</td>
<td>Can create copies of databases only</td>
</tr>
<tr>
<td>Creates application-aware, full snapshot backups</td>
<td>Creates crash-consistent copies</td>
</tr>
<tr>
<td>Can apply differential or transaction log backups at restore time</td>
<td>Can only attach the database. Cannot restore it and apply log backups</td>
</tr>
<tr>
<td>Can back up databases in a consistent manner (all databases freeze at the same time) on one server</td>
<td>Can create consistent copies of databases on several servers (federated databases). However, there are limitations. The Product Guide has more information.</td>
</tr>
<tr>
<td>On Windows 2003 (non-federated jobs), VSS is used to flush I/Os and create a shadow copy of the file system.</td>
<td>On Windows 2003 (non-federated jobs), VSS is used to flush I/Os and create a shadow copy of the file system.</td>
</tr>
</tbody>
</table>

The replicas that Replication Manager creates using the VDI snapshot API or consistent split are copies of the volumes containing the production data. A volume is the smallest unit that can be captured within a replica and is the level at which restore occurs in Replication Manager. A volume should contain files from only one database, if possible. System databases should not be on the same volume as user databases. There are special considerations when recovering system databases and VDI does not support restoring them. If a volume contains more than one database and you need to do a restore, Refer to “How to run rmsqlrestore” on page 18.

When creating a replica using just an array’s consistent split technology, Replication Manager makes a crash-consistent copy of the entire database. Since VDI was not used, SQL Server is unaware that a copy of the database was made. This means you cannot restore log backups at recovery time. Instead of restoring and recovering the database, Replication Manager attaches the database. When Replication Manager
uses VDI to create the replica, the replica is not considered transactionally consistent. But because VDI was used and it provides the metadata needed to recover the database, the replica is application consistent. It can be made transactionally consistent by using the VDI metadata and applying logs as is normally done during restore.

An application set determines what will be included in a replica. Replication Manager supports creation of replicas for databases and filegroups. However, the consistent split option is not supported if a filegroup is selected. This is because recovery of data in a filegroup replica also requires recovery of transaction logs.

Consistent split is useful when you have federated databases. Replication Manager supports federated databases on one server (composite application set) and multiple servers (federated application set). If you have Oracle and SQL Server databases on CLARiiON storage on the same server, you can create a composite application set and use consistent split to create a consistent copy of the data across Replication Manager database agents.

**Creation of replicas using the VDI snapshot API**

Although SQL Server supports full, differential, and transaction log backups, the VDI snapshot API that Replication Manager uses supports full backups only. Replication Manager creates full backups of SQL Server databases or filegroups using the VDI snapshot API. A full database backup includes all of the data (including full text catalogs for SQL Server 2005) and log files. A full filegroup backup includes all of the files in the filegroup.

When using VDI to create a replica, Replication Manager sends a Transact-SQL `BACKUP` statement to SQL Server. The replica is considered a full database or filegroup backup by SQL Server and impacts the backup sequence of databases. There is one exception: SQL Server 2005 added support for copy-only backups, which do not affect the backup sequence.

---

Note: Filegroup backups are supported only for databases that have either the “Full” or “Bulk-logged” recovery model set.

---

The SQL Server agent uses the VDI snapshot API to prepare the database for replication (freeze I/O to the database) and to resume normal operation after the replica is created (thaw I/O to the database). In addition, Replication Manager uses the Microsoft ODBC API and, on Windows Server 2003, the Volume Shadow Copy Service (VSS). The SQL Server agent uses the ODBC API to send Transact-SQL (T-SQL) statements to SQL Server to discover information about the databases and to manage replication and restore. On Windows Server 2003, VSS is used to create a shadow copy of the volume. Replication Manager uses VSS to manage the hardware portion of replication to flush I/Os and create the copy of the volume using the selected EMC storage technology. The SQL Server VSS Writer is not used as it does not currently have all of the features that the VDI API has.
Mounting SQL Server replicas

Unless you are creating a replica for repurposing, EMC recommends that you do not recover a database on the mount server. Recovering the database causes the database to change. Restores of the database will usually fail once this happens. The replica can be restored as a file system and then the database can be attached, but no transaction log backups can be applied. If you need to recover the database to run a consistency check and you are using CLARiiON clones, you can create and mount a snap of the replica and the changes will be discarded when the replica is unmounted. Figure 3 shows how to create and mount a snap of the replica.

Figure 3. Create and mount a snap

Mounting for backup

In Replication Manager, it is possible to mount the database data and log files without recovering the database. Replication Manager simply mounts the volumes on which the database resides. On a mount, this is useful if the user wants to stream the data and logs to tape without impacting the production server.

Replication Manager provides an option to run a backup script after a replica has been mounted.

When backing up the replica to long-term storage, make sure you back up the VDI metadata file and the other metadata files that Replication Manager transfers to the mount host. They are needed if you need to recover the database.
**Mounting for reuse**

In addition to mounting a replica so it can be backed up to long-term storage, Replication Manager can mount a replica and recover the SQL Server database so it can be reused. If the consistent split option is used, Replication Manager can attach the database to an instance of SQL Server running on the mount host. Otherwise, Replication Manager can recover the database using one of these three options:

- **Recovery** – Instructs the recovery operation to roll back any uncommitted transactions. After the recovery process, the database is ready for use.

- **No Recovery** – Instructs the recovery operation not to roll back any uncommitted transactions. When in No Recovery mode, the database is unusable. This option is useful when the database administrator needs to restore one or more transaction log backups.

- **Standby** – Restores files and opens the database in read-only mode. Subsequently, the database administrator can manually apply additional transaction log backups taken after the mirror split.

When recovering a database on a mount host, the database can be moved to a different path from where the database exists on the production server and the database can be renamed. Replication Manager can recover to any instance of SQL Server that is running on the mount host. It should be the same version of SQL Server as the production host. If the mount host and the production host are the same server, you must recover the database to another instance of SQL Server and move the database files to a new location by selecting an alternate mount path.

**Restoring SQL Server databases**

This section describes the methods for restoring a SQL Server database.

**Restoring from the replica**

Before restoring a database, you should attempt to back up the transaction log first. This minimizes the amount of data that is lost. Replication Manager can back up the transaction log. You need to use one of the SQL Server management tools to restore it after the database or filegroup has been restored.

Replication Manager can perform two types of restore – a full database or a filegroup restore. A full database restores all of the files belonging to the database including the transaction log and full text catalog files. A filegroup can be restored from a replica that contains a full backup of the database or a filegroup replica. When restoring a filegroup, you must back up the transaction log first. This is so the data in the filegroup can be brought forward to the same point in time as the other filegroups in the database.

The Recovery, No Recovery, and Standby options described previously for a mount operation are also available when restoring the full database. When restoring a filegroup, No Recovery is the only option available.
Restoring one database to a volume with many databases

The rmsqlrestore utility lets you restore individual SQL Server databases from a tape backup or mounted replica without reverse-syncing the target device over the source device. It can restore a database, filegroup, or file. The utility can restore to the original database or to a new database. The SQL Server VDI metadata that was created as part of the replication activity is required to restore a database using rmsqlrestore.

Restoring an individual database from a mounted replica is especially useful when you need to recover only one database and do not want to overwrite an entire device, as would happen with a normal Replication Manager restore.

How to run rmsqlrestore

The rmsqlrestore utility is a command line interface that you run from a command prompt window on the Replication Manager client. rmsqlrestore is installed on the client as part of the product installation.

Steps for restoring with rmsqlrestore

The exact steps you need to take may differ from the following, but the basic steps are:

1. Log in to the SQL Server system as a user with Administrator rights.
2. Back up the SQL Server transaction log using the SQL Server management tools or using the Query Analyzer to run a command like:
   
   ```
   BACKUP LOG "MYDB" TO DISK = "S:\sql
dumps\mydb.trn"
   ```

3. If restoring a full database, take the target SQL Server database offline. For file or filegroup restore, the database must be online.
   
   ```
   ALTER DATABASE "MYDB" SET OFFLINE
   ```

4. If restoring a database, restore all of the database files (.ldf and .mdf) from tape, or copy them from a mounted replica. You can copy them over the original files or to a new location. rmsqlrestore supports moving a database to a new location when doing a full restore. It does not support moving during a file or filegroup restore.

5. Open a command prompt window and cd to:
   
   ```
   C:\Program Files\EMC\rm\client\bin
   ```

   or

   ```
   C:\Program Files (x86)\EMC\rm\client\bin
   ```

6. Run the rmsqlrestore command. Complete syntax and sample commands are given in the syntax section following. The basic command syntax is:
rmsqlrestore -s <SQLservername> -d <databasename> -f <metadat file> -r <recovery_type>

7. If necessary, apply transaction logs and recover the database using the SQL Server management tools or using the Query Analyzer to run a command like:
   
   RESTORE LOG "MYDB" FROM DISK "S:\sql dumps\mydb.trn"

**Performing the file or filegroup restore**

Be sure you understand how restores of files and filegroups work in SQL Server before proceeding. All of the transaction log backups that have been created since the filegroup backup will be required to recover the database. The files in the filegroup must be brought back to the same point in time as the other files in the database.

1. Log in to the SQL Server system as a user with Administrator rights.
2. Be sure the target SQL Server database is online.
3. Be sure that the transaction log is backed up.
4. Open a command prompt window and cd to:
   
   C:\Program Files\EMC\rm\client\bin
   or
   C:\Program Files (x86)\EMC\rm\client\bin

5. Run the rmsqlrestore command. To restore two files, for example, run:

   rmsqlrestore -s <SQLservername> -d <databasename> -f <metadatafile> -lf <logical_filename1> -lf <logical_filename2> -r norecovery

To restore two filegroups, run:

rmsqlrestore
-s <SQLservername>
-d <databasename>
-f <metadatafile>
-lf <logical_filename1>
-fg <logical_filegroupname1>
-fg <logical_filegroupname2>
-r norecovery

Do not use quiet mode for a file or filegroup restore. You can use -lf and -fg in the same restore command.

6. When rmsqlrestore displays the restore command that it is about to run, verify with Y if it is correct.

7. When rmsqlrestore prompts, restore the files you are recovering, enter Y to continue.
The `rmsqlrestore` command syntax

Table 2 lists the command options for `rmsqlrestore`.

### Table 2. `rmsqlrestore` command options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Required</strong></td>
<td></td>
</tr>
<tr>
<td><code>-s</code></td>
<td>SQL Server name including instance name (host\instance)</td>
</tr>
<tr>
<td><code>-f</code></td>
<td>Metadata file name and location</td>
</tr>
<tr>
<td><code>-d</code></td>
<td>Database name</td>
</tr>
<tr>
<td><strong>Connection types (-e or –U)</strong></td>
<td></td>
</tr>
<tr>
<td><code>-E</code></td>
<td>Trusted connection (default)</td>
</tr>
<tr>
<td><code>-U</code></td>
<td>SQL Server login ID</td>
</tr>
<tr>
<td><code>-P</code></td>
<td>Clear text password (used with <code>-U</code> option)</td>
</tr>
<tr>
<td><code>-p</code></td>
<td>Encrypted password (used with the <code>-U</code> option)</td>
</tr>
<tr>
<td><strong>Optional</strong></td>
<td></td>
</tr>
<tr>
<td><code>-r</code></td>
<td>Recover option - RECOVERY, NORECOVERY (default), or STANDBY</td>
</tr>
<tr>
<td><code>-u</code></td>
<td>Undo filename, required for STANDBY</td>
</tr>
<tr>
<td><code>-m</code></td>
<td>Move file. Option has two parameters: logical_file_name and operating_system_file_name Pathnames must exist. Repeat option for each file, including the log file. If you are restoring to a new database name, use the <code>-m</code> option so you do not overwrite the original files. For example: <code>-m logicalfilename S:\existingdir\newfilename.mdf</code></td>
</tr>
<tr>
<td><code>-fg</code></td>
<td>Filegroup to restore. Repeat option for each filegroup</td>
</tr>
<tr>
<td><code>-lf</code></td>
<td>Logical file to restore. Repeat option for each logical file</td>
</tr>
<tr>
<td><code>-e</code></td>
<td>Displays encrypted password. Not used with other parameters</td>
</tr>
<tr>
<td><code>-v</code></td>
<td>Verbose mode</td>
</tr>
<tr>
<td><code>-q</code></td>
<td>Quiet mode. Will not ask questions</td>
</tr>
<tr>
<td><code>-l &lt;log_dir&gt;</code></td>
<td>Creates log files in the specified directory</td>
</tr>
<tr>
<td><code>-h</code></td>
<td>Help</td>
</tr>
</tbody>
</table>

**Examples**

To restore without applying logs using a trusted connection:

```bash
rmsqlrestore -s sql1\instanc1 -d custinfo -f "C:\sqlmounts\RMSQLMETADATA_lrma047 test db on R" -r RECOVERY
```

To restore to a new database name and move files using a SQL login and encrypted password:

```bash
rmsqlrestore -s sql1\instanc1 -d custinfoTest -f "C:\sqlmounts\RMSQLMETADATA_lrma047 test db on R" -r RECOVERY -m custinfo_Data S:\custinfoTest.mdf
```
-m custinfo_Log T:\custinfoTest.ldf
-U sa -p 1EMC_4roJdyU5;x

To get the encrypted password:
rmsqlrestore -e unencrypted_password

Special Considerations

For SQL Server 2005 and higher in cluster configurations, the SQL Server Browser Service needs to be running when you create the first application set for a clustered SQL Server instance.

After the first application set is created, the SQL Server Browser Service does not need to be running. The connection information is stored in the registry key: \KEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\SuperSocketNetLib\LastConnect. You can create subsequent application sets and run jobs without the SQL Server Browser Service running as long as the information in the registry key is correct.

However, if dynamic ports are being used, it is best to leave the Browser service running as a restart or failover can change the port number.

Conclusion

Replication Manager dramatically simplifies the management of Microsoft SQL Server databases through the use of disk-based replicas to improve availability and rapid recovery of mission-critical data. Replicas can be automatically mounted to alternate servers for backup and other tasks, with no impact on the production server.

References

For more information about Replication Manager, refer to the following sources:

- *EMC Replication Manager Administrator’s Guide*
- *EMC Replication Manager Product Guide*
- *EMC Replication Manager Release Notes*
- *EMC Replication Manager Support Matrix* on E-Lab Interoperability Navigator