

# Surround SCM Backup and Disaster Recovery Solutions

Investing in a source code management application, like Surround SCM, protects your code from accidental overwrites, deleted versions, and other common errors. The next step in protecting your code is investing time and money in an effective backup and disaster recovery solution for Surround SCM. The challenge is to schedule backup and recovery activities so they minimally impact access to data.

This paper outlines several methods of protecting Surround SCM and the assets it contains. The method you choose depends on your Surround SCM configuration, the size of your organization and your budget.

## Choosing the right method

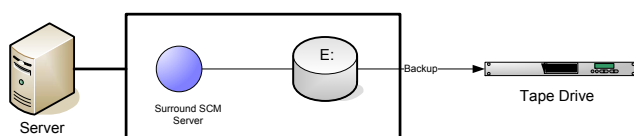
In Surround SCM 2009 and later, the server database and mainline branches are stored in an RDBMS database. The server and mainline database tables are stored in the same RDBMS database on the Surround SCM Server computer by default. The following backup methods use the default database configuration as an example, but you may be able to use the same concepts for distributed configurations.

If you have an RDBMS backup strategy, you may be able to use the existing processes to back up Surround SCM. If you are implementing a new backup strategy, consult any backup procedures and best practices provided by the RDBMS vendor. Your DBA can help you determine the best way to include Surround SCM databases in regular backups.

## Single server methods

### Back up to tape

The simplest way to back up your Surround SCM database is to connect a tape drive to the server.



Use the following steps to perform the backup.

1. Lock the database. Locking the database allows you to create backup copies without stopping the Surround SCM Server. It also prevents users from making any changes to the data during the backup.

Some configurations may require you to stop the Surround SCM Server and RDBMS server before performing a backup.

2. Back up the Surround SCM database on the E: drive to tape. (For simplicity, drive letters E: and F: are used throughout this paper.) The database location varies based on the RDBMS.
3. When the backup is complete, unlock the database or restart the Surround SCM Server and RDBMS server.

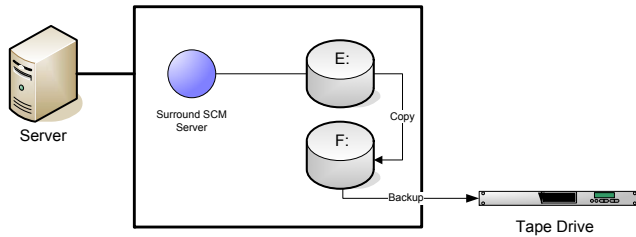
The entire process can be scheduled by using a script, Surround SCM's CLI commands, and a scheduling utility such as Cron or Windows Scheduled Tasks.

While this method is straightforward, it can be time consuming and the database may remain locked for long periods of time. The amount of time it takes to back up is related to the size and number of files in the Surround SCM database. This may not be an issue if all users are centrally located and you can back up the files overnight. If users are dispersed across several time zones or your database is so large that it cannot be backed up in the available time period, you should use one of the other methods discussed in this paper.

### Copy to an internal volume

If your external storage device is slow, you can possibly reduce Surround SCM downtime by copying the database to an internal volume and then backing up the copy.

The following scenario uses separate drives or volumes on the same server.



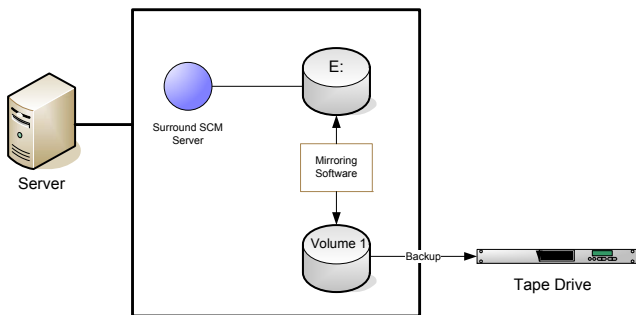
At a time when the Surround SCM Server has the least amount of traffic, use the following steps to back up the Surround SCM database.

1. Lock the database on the E: drive.
2. Copy the Surround SCM database from E: to F:.
3. Unlock the database on the E: drive.
4. Back up the F: drive to tape at your convenience.

You can use Robocopy, rsync, or a similar utility to copy between the E: and F: drives. Copy time is directly related to the speed of the hardware, the operating system, and the database size.

### Software mirroring

If you want to increase the regularity of backups without investing in an additional server, you can use an application like Veritas Volume Manager or Symantec's Storage Foundations for Windows to create and synchronize two mirrored drives.



For this type of solution, the underlying software layer should be set up before installing Surround SCM.

When you install the mirroring software, it creates an initial copy of the drive. You can then schedule the software to synchronize the mirror with another volume. The benefit of this method is that the mirroring software only copies the changes instead of copying the entire volume. Data is also copied at the block level instead of the file level, which provides a significant performance improvement.

After the mirror is established and the synchronization process is scheduled, use the following steps to back up the data.

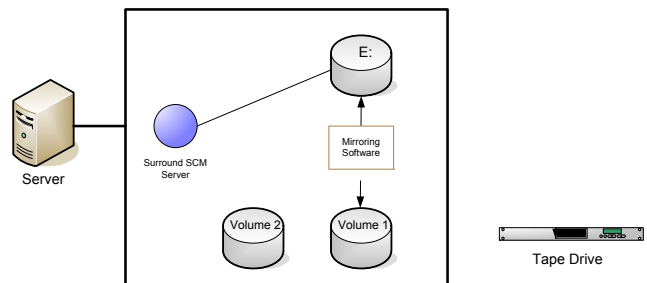
1. Lock the database on the E: drive.
2. Break the mirror between the two drives so that the software does not try to synchronize with the E: drive while you are backing up the mirrored volume.
3. Unlock the E: drive.
4. Add a volume name for the mirror (for example, F:).
5. Back up the mirrored volume to tape.
6. When the back up is complete, detach F:.
7. Reattach the mirror with the E: drive.

With this method, the downtime for users is minimal because the database is only locked for as long as it takes to break the mirror.

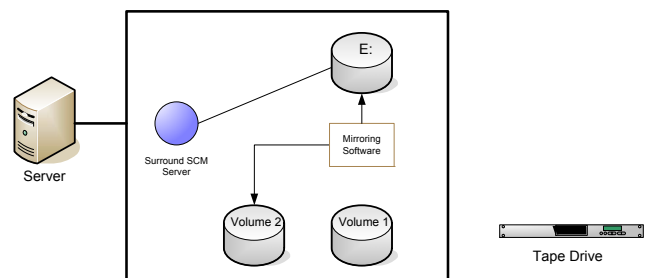
### Software mirroring with multiple drives

In the previous example, there is a primary volume and a mirrored volume. When you break the mirror between the two drives, you run the risk of losing changes while the mirror is backed up. For example, suppose that you break the mirror at 10:00 p.m. and it takes four hours to back up the mirrored drive to tape. If corruption occurs on the E: drive at 1:30 a.m., you will have to restore from the 10:00 p.m. copy and will lose three and a half hours of changes because no mirroring occurred during the backup.

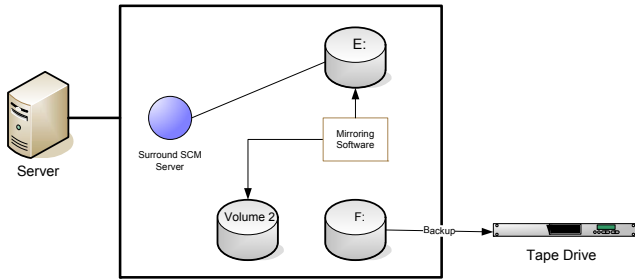
You can reduce the likelihood of losing changes while the mirror is being backed up by using two mirrored drives in the following arrangement.



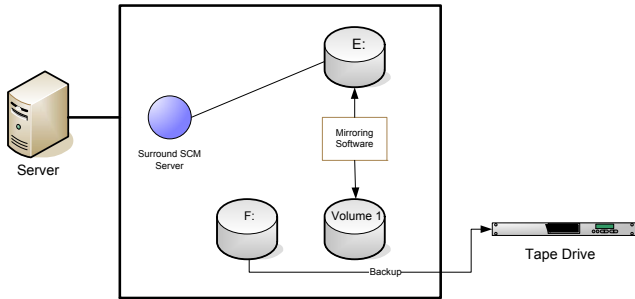
To use two mirror volumes, first lock the database on the E: drive and break the mirror for Volume 1. Then attach the mirroring software to the second volume.



The mirroring software will resynchronize the data with Volume 2. While Volume 2 is attached, simply mount Volume 1 and back up that data to tape.



After the backup is complete, detach F: (Volume 1). When you are ready for the next back up, break the mirror for Volume 2 and re-establish it for Volume 1.



You can continue the process of rotating mirrored volumes as you perform backups to ensure that users always have access to current data.

## SAN options

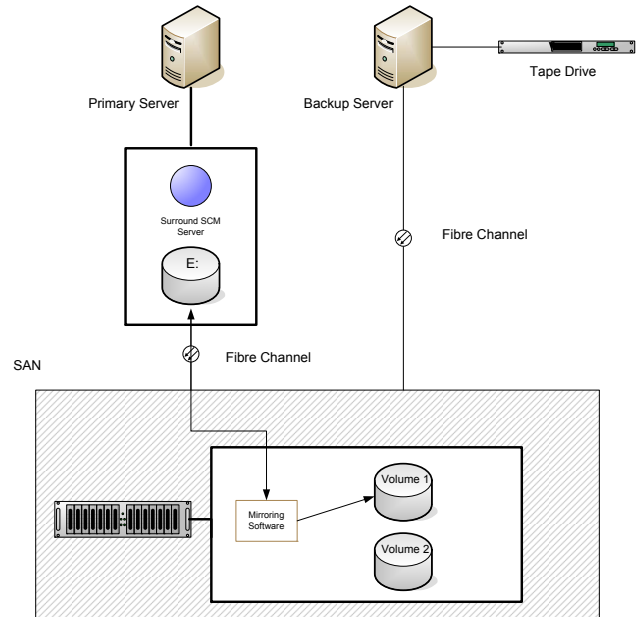
All of the examples presented so far can be implemented with a single server and are appropriate for smaller companies with centrally located employees and a limited hardware budget. Larger companies with multiple locations require a Storage Area Network (SAN).

### Local SAN

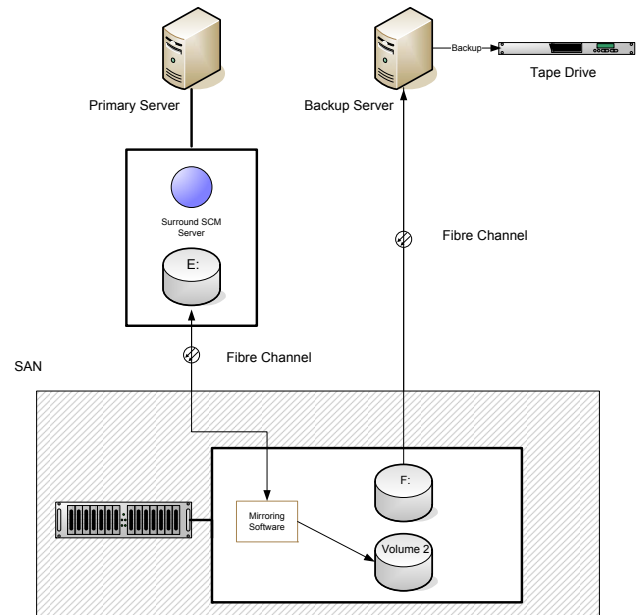
The basic principle behind the local SAN is the same as the software mirroring method. The following configuration builds off the method that mirrors to multiple drives.

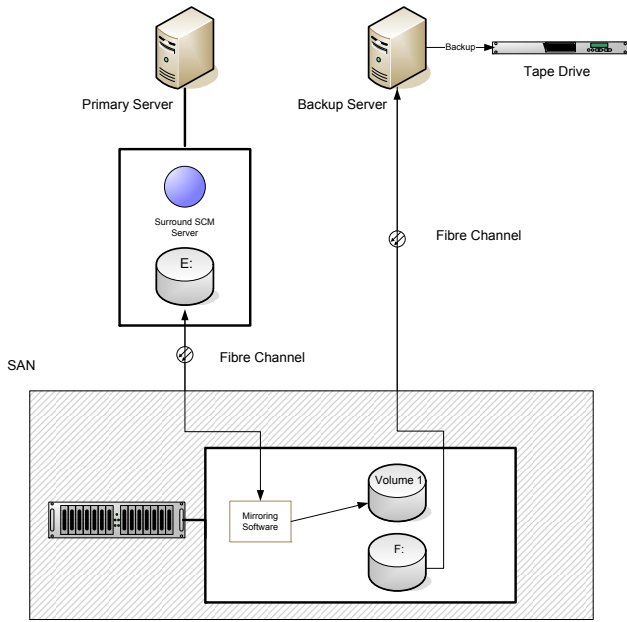
The backup procedure is similar to the one used with a single server.

1. Lock the database on the E: drive of the primary server.
2. Break the mirror.
3. Unlock the database on the E: drive.
4. Add the volume for the mirror on the SAN (F:).
5. Attach F: to the backup server.
6. Attach the mirroring software to the second volume.



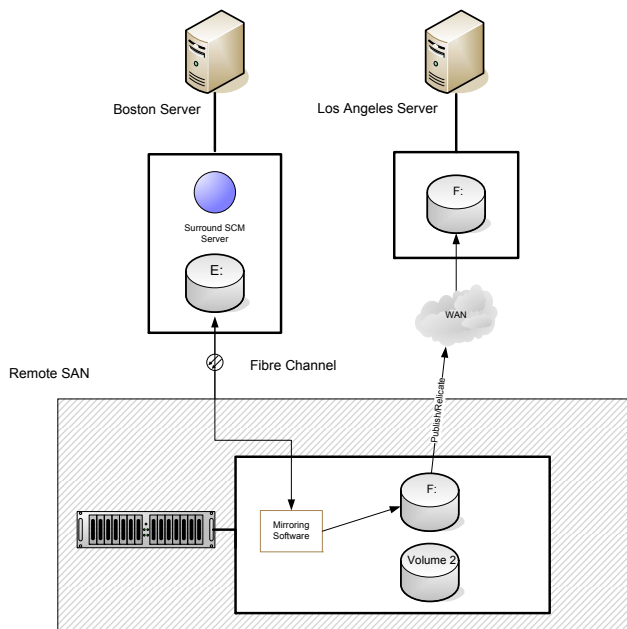
After the F: drive is attached to the backup server, you can back it up to tape. When the backup is complete, detach the F: drive from the backup server and re-establish the mirror. You can then name the second volume and attach it to the backup server.





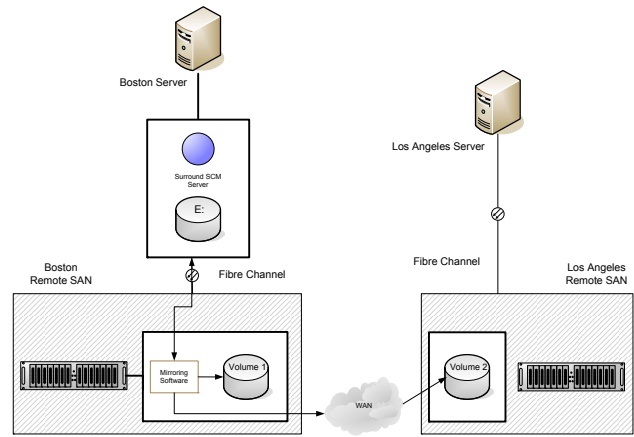
### SAN with remote publication or replication

If you have multiple locations, you may want to publish the mirrored volume to a remote location to protect against a disaster. In the following example, a company has offices in Boston and Los Angeles. All users access Surround SCM using the Boston server. The Surround SCM database is mirrored to a volume on the Boston SAN and that mirrored volume is replicated over a wide area network (WAN) to the Los Angeles server, which acts as a remote backup server.



Each SAN manufacturer has different methods for replication and publication. Consult with your SAN vendor for the solution that is most appropriate for your environment.

To keep the backup in Los Angeles more up to date, you can create mirrored volumes on two different SANs, one in Boston and one in Los Angeles.

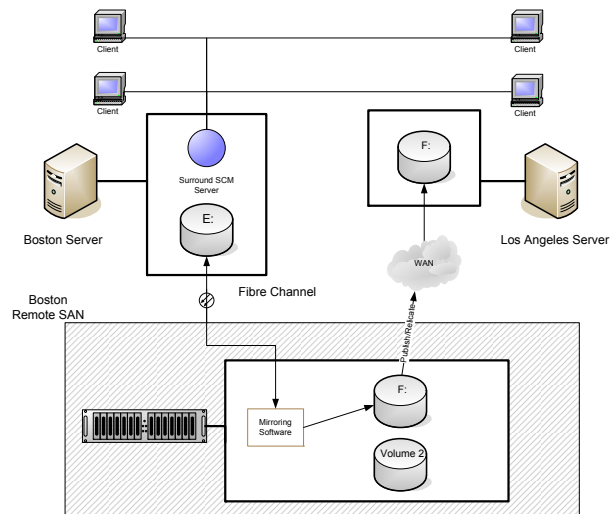


In this case, however, Volume 1 and Volume 2 will not be completely synchronized because mirroring is performed over a WAN.

*(Seapine validated these configurations using an EMC CLARiiON CX3 model 80.)*

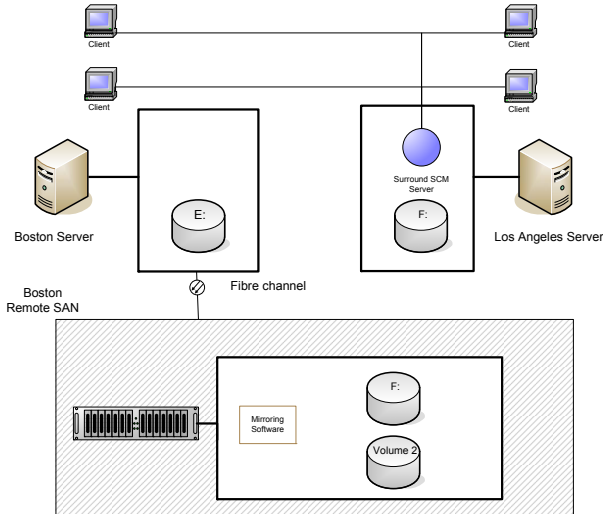
### Remote disaster recovery

The SAN with remote publication or replication method provides the best disaster recovery solution because the mirrored copy of the database is stored on separate hardware in a remote location. Building on that, we will consider the Surround SCM recovery process, which is a manual process and not an automatic failover routine. The following diagram shows an example of a remote SAN environment.



All users connect to the Boston server, which houses the Surround SCM Server and database. There are mirrored copies of the database on the Boston server and the Los Angeles server.

Suppose that the Boston server experiences a fatal crash. To recover the data, the Surround SCM Server is installed on the Los Angeles server and configured to point at the local mirrored copy of the database. The users in both locations then configure their clients to use the Surround SCM Server in Los Angeles.



If the Boston server is restored or a new server implemented, the database can be moved from Los Angeles back to Boston. Users would configure their clients to access the Boston server again.

To minimize manual configuration, client computers can also access the server via a name instead of a specific IP address. In the cutover situation, you can modify the DNS server to have the name point to the new Surround SCM Server instead of requiring users to configure their Surround SCM client settings.

## Conclusion

You can greatly improve your disaster recovery efforts using any of the methods outlined in this paper. Your goal should be to make your architecture as redundant as your budget allows. The solution you choose depends on the size of your organization, the backup frequency you require, and your hardware budget.