Backup Agent

For Microsoft SQL

Option Guide

COMPUTER® ASSOCIATES
Software superior by design.

This documentation and related computer software program (hereinafter referred to as the "Documentation") is for the end user's informational purposes only and is subject to change or withdrawal by Computer Associates International, Inc. ("CA") at any time.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO THE END USER OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, BUSINESS INTERRUPTION, GOODWILL OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED OF SUCH LOSS OR DAMAGE.

THE USE OF ANY PRODUCT REFERENCED IN THIS DOCUMENTATION AND THIS DOCUMENTATION IS GOVERNED BY THE END USER'S APPLICABLE LICENSE AGREEMENT.

The manufacturer of this documentation is Computer Associates International, Inc.

Provided with "Restricted Rights" as set forth in 48 C.F.R. Section 12.212, 48 C.F.R. Sections 52.227-19(c)(1) and (2) or DFARS Section 252.227.7013(c)(1)(ii) or applicable successor provisions.

All product names referenced herein belong to their respective companies.

Contents

Chapter 1: Introduction

Chapter 2: Using the Backup Agent with MicroSoft SQL Version 6.5

Chapter 3: Using the Backup Agent with Microsoft SQL Version 7.0 and above

Appendix A: Troubleshooting and Disaster Recovery

Chapter

1 Introduction

The Backup Agent for Microsoft SQL is a "go-between" that allows ARCserve to efficiently back up and restore SQL Server databases. The Backup Agent helps ARCserve retrieve and send data to and from the SQL Server in a consistent manner.

Using ARCserve and the Backup Agent, you can:

- Remotely manage backups of SQL Server databases.
- Perform online database backups by taking advantage of the SQL Server dynamic dump utility.
- Schedule your backup.
- Back up to a wide array of tape storage devices.

How the Backup Agent Works

The Backup Agent resides on the server with the SQL Server. When ARCserve starts to back up a database, it sends a request to the Backup Agent. The Backup Agent retrieves the database from the SQL Server (using the SQL Server DB-Lib functions), and sends it to ARCserve, where the complete database is backed up to tape. Similarly, the Backup Agent transfers the database information when the database is restored from tape.

The Backup Agent takes advantage of the SQL Server's Dump Database/Transaction backup method. This method backs up the database and/or transaction log in a single step. The Backup Agent ensures that a complete database is backed up; transaction logs are backed up with the database by default.

For each database that is backed up, Backup Agent initiates a "dump" in SQL Server. The SQL Server "sends" the database in multiple data chunks; The Backup Agent receives one data chunk at a time, and passes it directly to ARCserve, where it is backed up to a tape drive.

The following steps describe the data flow when ARCserve uses Backup Agent to back up an SQL Server database:

1. ARCserve sends a request to the Backup Agent for a database.
2. The Backup Agent instructs the SQL Server to perform a backup of a particular database or log.
3. The SQL Server returns the data from the database in multiple "chunks" to the Backup Agent, one chunk at a time.
4. The Backup Agent retrieves the data chunks from the SQL Server and transfers them to ARCserve.
5. ARCserve writes the data chunks to tape or disk, depending on the user's backup destination selection.
6. Steps 2 through 5 are repeated until there is no more data to be backed up.

Backup Agent and SQL Server's Backup function guarantees consistency and accuracy of the data being backed up.

Microsoft SQL 2000

When you use the Backup Agent on a Microsoft SQL 2000 server, the following additional features are available:

- Multiple instance support— Backup and restore support on multiple instances of the SQL server running concurrently on the same computer.
- Restore Options— Restore with stopping to a specific mark, restore with stopping before a specific mark, restore with keep replication settings, restore with restricted user access, partial restore, and allowing of the moving of log files. Refer to chapter 3 for further information on these options.

Multiple Instance Support

Microsoft SQL Server 2000 supports multiple instances of the SQL Server running concurrently on the same computer, each instance having its own set of system and user databases that are not shared between instances. Applications can connect to each SQL Server instance on a computer in the same way they connect to SQL Server running on different computers.

The Backup Agent offers backup and restore support for multiple SQL Server instances. The ARCserve Backup Manager display instances for the local computer and for the remote computer, one of them being expanded; the default instance is Microsoft SQL Server.

Requirements

ARCserve 2000 version 7.0 can be installed on the following platforms:

Operating System	Windows NT 4.0 with Service Pack 5	or	Windows 2000 Advanced Server
Database Version	Microsoft SQL Server v6.5 Or Microsoft SQL Server v7.0 with Service Pack 1 (Build 7.00.699)	or	Microsoft SQL Server v7.0 with Service Pack 2 (Build 7.00.842) or Microsoft SQL Server 2000

Chapter 2

Using the Backup Agent with MicroSoft SQL Version 6.5

Backup Overview

To *back up* or *dump* is to create a copy of a database or transaction log on another device (usually a tape drive). Backups are accomplished using ARCserve, the Backup Agent, and the SQL DUMP statement. Backing up a database backs up both the database and the transaction log.

Databases

Backing up a database creates a copy of its tables, data, and user-defined objects. In case of media failure, you can recover your databases if you have been making regular backups of your databases and their transaction logs. Transaction logs are areas reserved by SQL Server to record changes that are made to a database. Each database has its own transaction log. Every change made to a database is automatically recorded in its transaction log.

Transaction Logs

The transaction log records data modification requests (INSERT, UPDATE, or DELETE statements) as they are executed. When a transaction begins, a begin transaction event is recorded in the log. This event is used during recovery to determine the starting point of a transaction. As each data modification statement is received, it is recorded in the log. The change is always recorded in the log before that change is made in the database itself. This type of log is called a write-ahead log.

The transaction log is shared by all users of the database. Multiple changes are frequently recorded each time a log page is written to the database device. This greatly improves I/O efficiency.

When an SQL database backup is started in ARCserve, the Backup Agent initiates a dynamic dump of the database. The dump can take place while the database is active. The dump captures the state of the data as it is at the moment the statement is executed; no partial transactions are reflected. Any data changes made after the dump begins are not reflected in the dumped database. Performing a backup can slow the system down, so you may want to run it when the database is not being heavily updated.

ARCserve will indicate if a database has a separate storage space for its transaction log. In this case, you can back up the transaction log separately. Dumps of the transaction log are usually coordinated with database dumps as part of an overall backup procedure. A transaction log dump, like a database dump, can take place while the database device is active.

Typically, transaction logs are backed up more often than databases. A transaction log might be backed up once a day, and a database might be backed up once a week. Backing up a transaction log takes less time and uses less storage space than backing up a database. (Do not back up a transaction log unless the database has already been backed up at least once.)

It takes longer to recover a database when you restore the database and all transaction logs sequentially than it does when you restore only the database. Finding the correct strategy depends on your environment. The time required to perform backups must be considered in relation to the time required to restore.

In general, you should back up a database immediately after you create it, and then maintain a regular backup schedule to ensure smooth recovery in case of database or media failure. ARCserve allows you to set and maintain a schedule of automatic backups. Refer to the online *ARCserve Administrator Guide* for information on scheduling backups.

You should maintain regular backups of all databases, including:

- The master database
- The msdb database
- All user databases
- The distribution database (if the server is configured as a replication distributor)

Backing Up

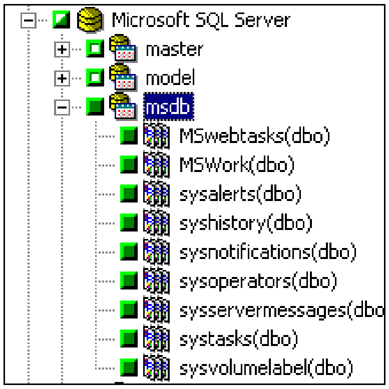
With small databases, the transaction log is usually stored on the same database device as the rest of the database. After backing up a small database, you should truncate the database log to remove committed transactions from the log. In large databases, the transaction log and the data portion of the database are usually stored on different devices. In that case, you can back up either the transaction log or the database.

1. Make sure that the SQL Server is running on your server. The MSSQL Server service must be started.
2. Start ARCserve and open the Backup Manager. Refer to the online *ARCserve Administrator Guide* for details on starting ARCserve. The Backup Agent is started automatically when you start ARCserve.

3. From the Source tab, locate the Microsoft SQL Server. It is listed under the server on which it is installed.

To Backup a Database

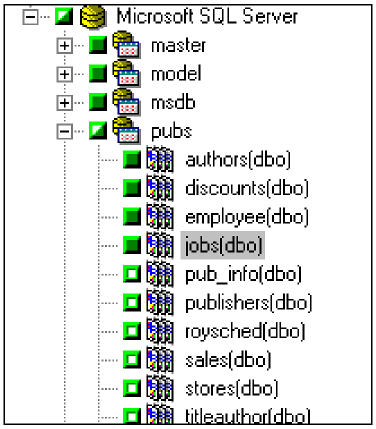
4. Double-click on the SQL Server to display a list of databases for this server and then select the database you wish to back up.



Note: The green box by the database name should be completely filled to perform a database level backup.

To Backup an Individual Table(s)

Double-click on the SQL Server to display a list of databases for this server and then expand the database to choose the individual tables you wish to back up.

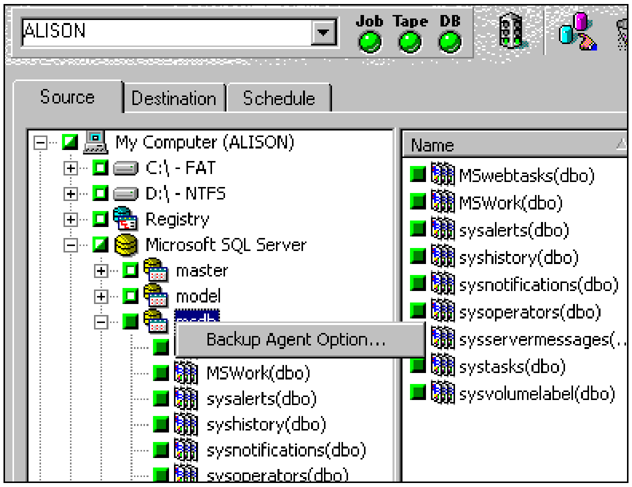


Note: When you select a table or combination of tables to be backed up, ARCserve treats each table as a separate session. The green box by the database name should be half full to perform a table level backup.

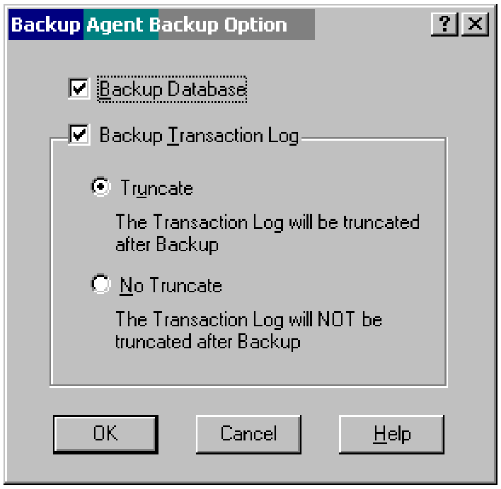
To Backup the Transaction Log

If the field LogSize is displayed, then this database has a separate transaction log. The transaction log is backed up with the database by default.

5. Right-click on the Backup Agent Option and a pop-up menu appears:



6. Select *Backup Agent Option* from the pop-up menu. The Backup Agent Backup Option dialog box opens. Note that Backup Agent options are not applicable for table level backups.


Backup Agent Backup Option

☑ Backup Database

☑ Backup Transaction Log

● Truncate

The Transaction Log will be truncated after Backup

○ No Truncate

The Transaction Log will NOT be truncated after Backup

7. Select your backup options and click OK.

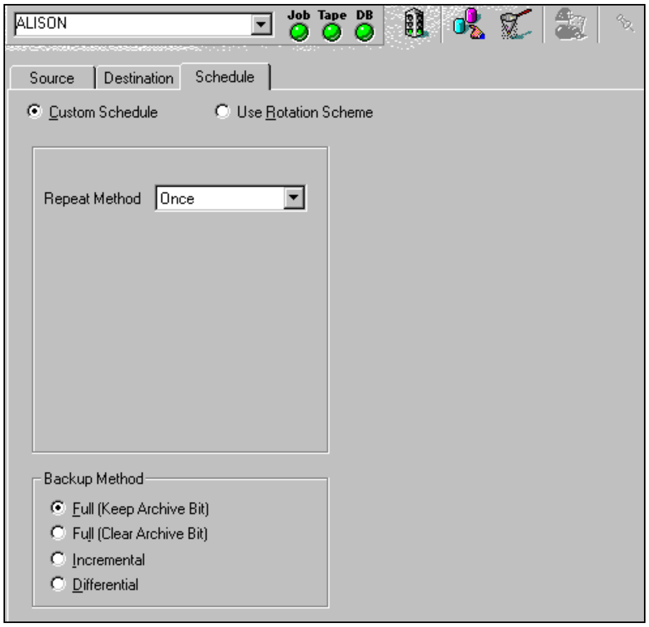
   ARCserve provides the following options when you back up a transaction log:

   - Truncate: Truncate the transaction log files after backup.
   - No Truncate: Do not truncate the transaction logs after backup.

   Truncating a transaction log removes the inactive portion of a log. You usually truncate a transaction log after you have backed up a database and transaction log together.

8. Click the Destination tab to select a backup destination.

9. Click the Schedule tab. The Schedule dialog appears. From this dialog, you can schedule the job and specify the backup method. To run the job now, you may bypass the schedule setup and go to the Run/Schedule button.



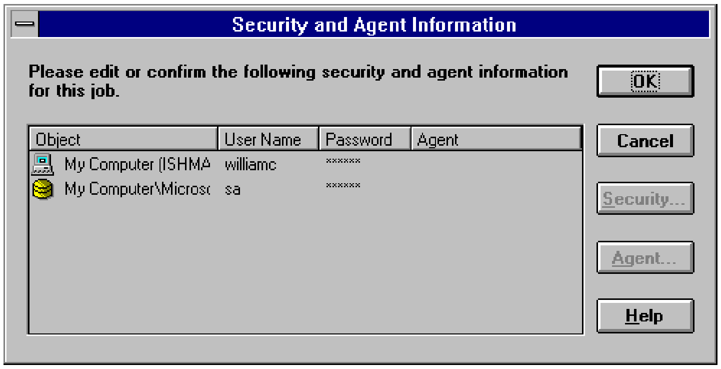
● Custom Schedule ○ Use Rotation Scheme

Repeat Method: Once

Backup Method

● Full (Keep Archive Bit)
○ Full (Clear Archive Bit)
○ Incremental
○ Differential

10. Click the Run/Schedule button on the toolbar. The Security and Agent Information dialog box opens.



Security and Agent Information

Please edit or confirm the following security and agent information for this job.

Object	User Name	Password	Agent
My Computer (ISHMA	williamc	******	
My Computer\Micros(	sa	******	

Note: In this dialog box, the column and button labeled "Agent" refers to the Client Agent, not the Backup Agent. If you wish, you may enter client agent information at this time.

11. Enter or verify the user name and password for the SQL Server. To enter or change security information for the SQL Server, select the SQL Server object and click Security; enter the new user name and password in the Security dialog box, then click OK.

12. After verifying or changing the security information, click OK to continue.

13. The Submit Job dialog box appears. Use this window to select the job execution time, to place the job on hold, to submit an optional description for the backup job, or to select source priority.

Refer to the online *ARCserve Administrator Guide* for further information on scheduling backups.

Restore Overview

To *restore* or *load* is to load a database from a backup of that database and (if applicable) one or more backups of its transaction log. A load overwrites any information in the database with the backed-up information. Loads are accomplished by using ARCserve, the Backup Agent, and the SQL LOAD statement.

If a database or table is damaged or lost, you can restore the database or table by reloading the most recent database or table backup and the succeeding transaction log backups. Or, if you have some corrupt data on the database or table, you can reload a database or table backup over the corrupt database or table.

If you are reloading a database or table because of media failure, you must do the following:

- Drop the damaged database or table using either of these SQL Server utilities: DBCC or the sp_dbremove system stored procedure.
- Recreate the database or table.
- Restore the database or table from backup.

You can also dump a database and load that dump into another database. However, the new database must be at least as large as the dumped database that is being loaded. (Segment mappings must also be the same.)

Note: If the msdb database is to be restored from a backup, the SQLExecutive service must be stopped before the restore begins and then restarted after the restore is complete. If this is not done, then SQL Executive will not function properly.

The Backup Agent enables the following possible Restore operations:

- Restore to original location (same server) with same database name
- Restore to original location (same server) with different database name
- Restore to different location (new server) with same database name
- Restore to different location (new server) with different database name

Database Loads

When a database is loaded, SQL Server rolls back any uncommitted transactions that were active at the moment the dump began. When the load is complete, the database is in the same state it was in when the DUMP statement was executed, minus any transactions that were active at that point.

Once the data from the dump has been reloaded, SQL Server reinitializes any remaining unused pages. For example, if a 100-MB database contains only 5 MB of data, all 100 MB of space is still rewritten. A database load command takes at least as long as the CREATE DATABASE statement that created the database.

Loading a database locks it so that it cannot be modified while recovery is in progress. Users can, however, access and modify other databases on SQL Server during this time.

If a failure occurs while a database is being loaded, SQL Server notifies the system administrator but does not recover the partially loaded database. The database load must be restarted.

The destination database must have at least as much storage space as was allocated to the dumped database. The actual amount of data in the dumped database is irrelevant. To get information about allocated storage space, use SQL Enterprise Manager or the DBCC CHECKALLOC statement.

After a media failure, restart SQL Server. If SQL Server cannot access a database, it marks the database as suspect, locks it, and displays a warning message. A damaged database must be dropped, which can be done using the SQL Enterprise Manager.

After the database is dropped, it must be recreated. After it is recreated, the system administrator or database owner can load the most recent database dump (plus any transaction log dumps) using ARCserve and the Backup Agent. (The master database is restored using a special procedure.)

Transaction Log Loads

Once you’ve restored a database, you can load the transaction log dumps that were done after the database dump. Loading transaction logs allows you to recover as much of a database as possible in case of system failure.

Backups of the transaction log must be loaded in the sequence in which they were made. SQL Server checks the timestamps on each dumped database and each dumped transaction log to see that the sequence is correct.

Loading a transaction log is also referred to as “applying” a transaction log. Loading a transaction log results in re-execution of the changes it contains and in rolling back any transactions that were uncommitted when the transaction log was backed up.

When entire sequence of transaction log dumps has been loaded, the database is restored to its state at the time of the last transaction log dump, minus active transactions. (The exception to this is the SQL DUMP TRANSACTION statement, which was active at that point. It is not rolled back. It is completed by the next LOAD TRANSACTION statement.)

Restoring a Master Database

Restoring the master database is a special case. You will need to rebuild the master database from the SQL server setup. (Refer to your Microsoft SQL Server documentation for complete instructions). Start SQL server in single-user mode, then restore the master from the most recent backup.

1. Stop SQLExecutive service.
2. Start MSSQL server in single user mode with the following command:
   ```
   sqlservr /c /dPATH_TO_MASTER_DEVICE /m
   ```
   Where PATH_TO_MASTER_DEVICE is the complete physical path to master.dat
3. After restore of master database, the physical name of the master device may appear as " the_master_device". This might cause subsequent problems in the operation of the Backup Agent.

   To update the physical device name of the master database in the sysdevices table in master database, run the following commands from ISQLW:
   ```
   use master
   go
   sp_configure "allow updates",1
   go
   reconfigure with override
   go
   begin trans
   go
   update sysdevices set phyname='c:\mssql\data\master.dat'
   where name='master'
   go
   commit trans
   sp_configure "allow updates",0
   go
   reconfigure with override
   go
   ```
   An error in this statement could directly affect the functionality of the SQL server. For more information please refer to the sp_configure procedure in the *Transact-SQL Reference Manual*.

Note: When restoring the master database, ARCserve will report the job as crashed although it is not. The reason is that the SQL Server unloads itself as soon as the restore of the master database is completed and the ARCserve restore job process is abruptly terminated. See Appendix A for Disaster Recovery.

Restoring

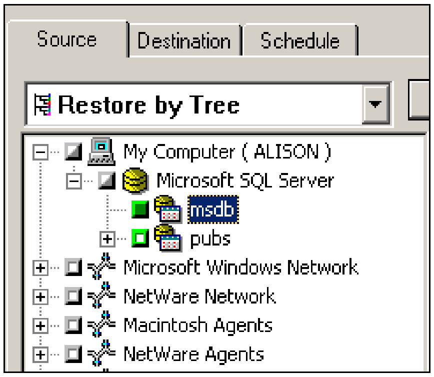
When a database or table is restored from a backup, that database or table must not be in use. Any data in the specified database or table is replaced by the loaded data.

To Restore a Database, Transaction Log or Individual Table:

1. Start ARCserve and open the Restore Manager. Refer to the online *ARCserve Administrator Guide* for details on starting ARCserve.

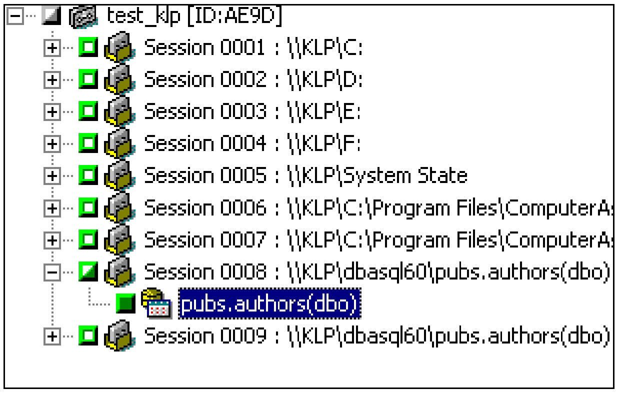
2. From the Source tab, select the *Restore by Tree* option.

3. In the Browser, locate the Microsoft SQL Server, then select the database or table you want to restore.



Individual Table

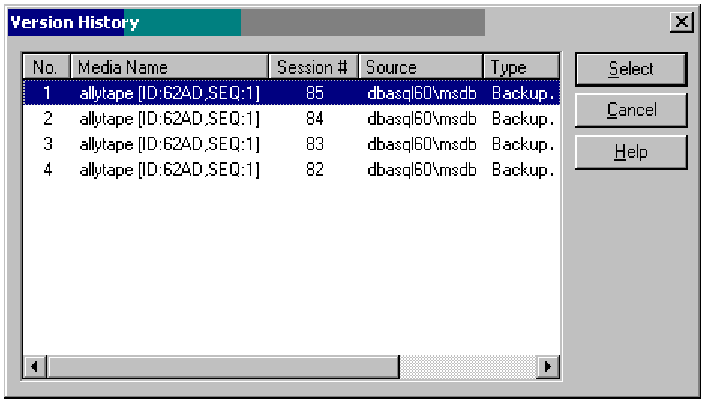
The table names will appear appended to the database name as such: `DATABASENAME.TABLENAME`


test_klp [ID:AE9D]
Session 0001 : \\KLP\C:
Session 0002 : \\KLP\D:
Session 0003 : \\KLP\E:
Session 0004 : \\KLP\F:
Session 0005 : \\KLP\System State
Session 0006 : \\KLP\C:\Program Files\ComputerA
Session 0007 : \\KLP\C:\Program Files\ComputerA
Session 0008 : \\KLP\dbasql60\pubs.authors(dbo)
pubs.authors(dbo)
Session 0009 : \\KLP\dbasql60\pubs.authors(dbo)

Transaction Log

The transaction logs need to be restored in the order they were backed up. If you use the Truncate option, you must restore in sequence 1, 2, 3, and so on. If you use the No Truncate option, you can restore the last log that was backed up.

4. Click the Version History button. The Version History dialog box opens:



No.	Media Name	Session #	Source	Type
1	allytape [ID:62AD,SEQ:1]	85	dbasql60\msdb	Backup.
2	allytape [ID:62AD,SEQ:1]	84	dbasql60\msdb	Backup.
3	allytape [ID:62AD,SEQ:1]	83	dbasql60\msdb	Backup.
4	allytape [ID:62AD,SEQ:1]	82	dbasql60\msdb	Backup.

5. Select a backup version for this database or table and click the Select button. The Version History dialog box closes.

6. Click the destination tab and select a destination.

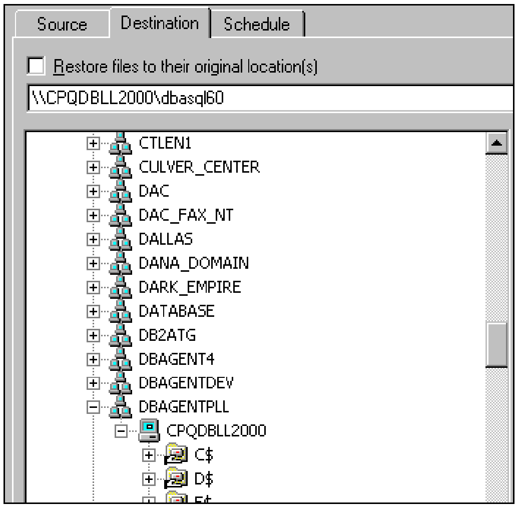
   The type of Restore operation that will be performed depends on the destination you select. You can select to restore to either the original server that was backed up, or to a different server. In either of these cases, you can restore the database with the same name as the original, or with a different name. See the appropriate procedures below:

To Restore to a Different Server using the Same Database Name

In the Browser, clear the “Restore files to their original location(s)” check box. Select the server you want to be the destination.

To Restore to a Different Server Using a Different Database Name

In the Browser, clear the “Restore files to their original location(s)” check box. Select the server you want to be the destination. Select the Microsoft SQL Server icon. In the space below the check box, enter the new name of the database at the end of the path displayed.



□ Restore files to their original location(s)

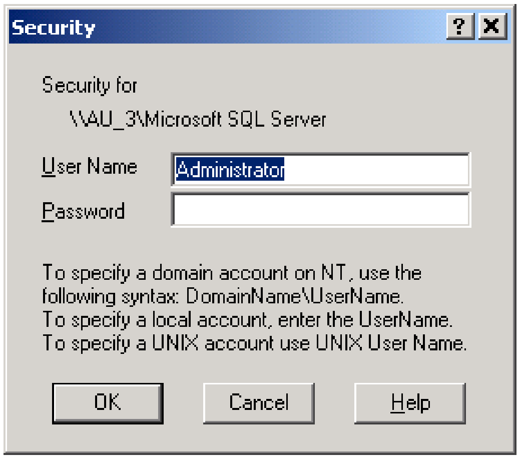
\\CPQDBLL2000\dbasql60

7. Click the Run/Schedule button on the toolbar.

8. Specify the Security Information according to the type of Restore operation you have selected.

To Specify Security When Restoring to the Original Server

Enter the username and password for your SQL Server. Click OK to continue.



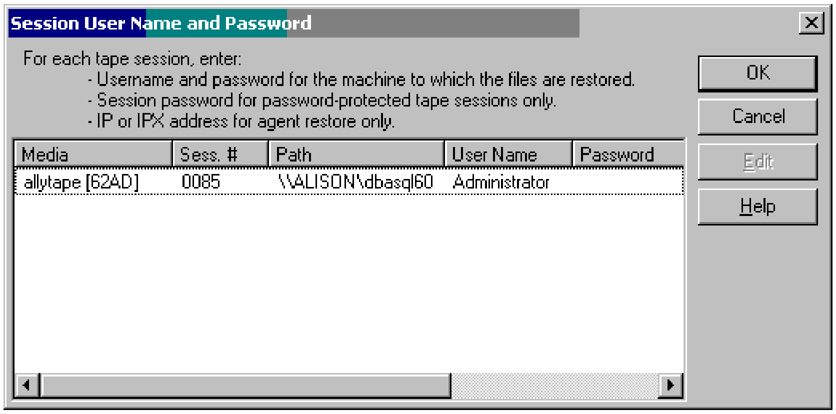
Security for
\\AU_3\Microsoft SQL Server

User Name: Administrator

Password:

To specify a domain account on NT, use the following syntax: DomainName\UserName.
To specify a local account, enter the UserName.
To specify a UNIX account use UNIX User Name.

The Session user name and password dialog box opens. Verify or change the user name and password for the Windows NT/2000 machine on which the SQL Server is loaded. To change the user name or password, select the Session and click Edit. If a Session password was assigned to this Session, enter the Session password.

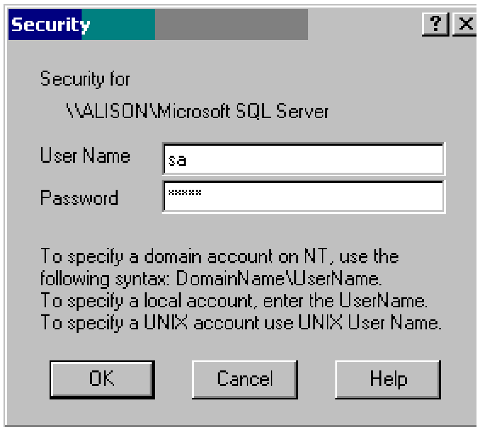


For each tape session, enter:
- Username and password for the machine to which the files are restored.
- Session password for password-protected tape sessions only.
- IP or IPX address for agent restore only.

Media	Sess. #	Path	User Name	Password
allytape [62AD]	0085	\\ALISON\dbasql60	Administrator	

To Specify Security When Restoring to a Different Server

Enter the user name and password for the Windows NT/2000 server and click OK. When the Security dialog box for the SQL Server appears, enter the user name and password for your SQL Server.

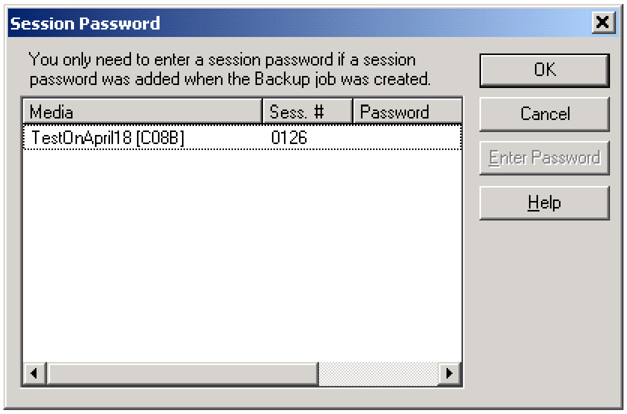


Security for
\\ALISON\Microsoft SQL Server

User Name: sa
Password: ******

To specify a domain account on NT, use the following syntax: DomainName\UserName.
To specify a local account, enter the UserName.
To specify a UNIX account use UNIX User Name.

Click OK to continue. The Session Password dialog opens. If a session password was assigned to this session, enter the session password. Click OK.



You only need to enter a session password if a session password was added when the Backup job was created.

Media	Sess. #	Password
TestOnApril18 [C08B]	0126	

Backing up and Restoring in a SQL Server Cluster Environment

ARCserve supports the Active/Passive configuration of the SQL Virtual Server in the Microsoft Cluster Server (MSCS) environment.

An SQL Virtual Server is backed up and restored just like any non-clustered SQL Server using the Backup Agent with the following important differences:

- The Backup Agent must be set up on the local drives of all nodes in the SQL Virtual Server.
- The SQL Virtual Server is always remote. The SQL Virtual Server is listed under the MSCS icon. Therefore it must be backed up through the network or the preferred shares under the MSCS icon in the Backup Manager. It should not be backed up through the local machine or the SQL Virtual Server icon.
- If the node that the SQL Virtual Server is currently running in fail over during a backup job, the back up job will fail and must be resubmitted.

Chapter

3

Using the Backup Agent with Microsoft SQL Version 7.0 and above

Backup Overview

To back up is to create a copy of a database, transaction log, differential backup, or file/file group on another device (usually a tape drive). Backups are accomplished using ARCserve, the Backup Agent, and the SQL BACKUP statement.

Backing up a database creates a copy of its tables, data, and user-defined objects. In case of media failure, you can recover your databases if you have been making regular backups of your databases and their transaction logs.

When an SQL database backup is started in ARCserve, the Backup Agent initiates a dynamic backup of the database, a backup that can take place while the database is active. The backup captures the state of the data as it is at the moment the statement is executed; no partial transactions are reflected. Any data changes made after the backup begins are not reflected in the backed up database.

Ideally, you should back up a database immediately after you create it. Thereafter, you should continue to back it up on a regular schedule to ensure smooth recovery in case of database or media failure. You should maintain regular backups of all databases, including:

- The master database
- The msdb database
- All user databases
- The distribution database (if the server is configured as a replication distributor)

ARCserve allows you to set and maintain a schedule of automatic backups. Refer to the online *ARCserve Administrator Guide* for information on scheduling backups.

Tip: Performing a backup can slow the system down, so you may want to run it when the database is not being heavily updated.

A recommended approach is:

1. Back up the database regularly, at least once every night.
2. Perform differential backups frequently, like every five hours or more for highly active databases.
3. Perform transaction log backups between each differential backup every 30 minutes.

Differential Backups

You can significantly reduce your backup time by supplementing your full database backups with a differential backup. A differential backup records only those data changes made to the database after the last full database backup. A differential backup is smaller and takes less time to complete.

If your system has some of the following characteristics, you may want to consider differential backups:

- You do not often have the resources (such as lack of storage space or lack of time) to perform full database backups.
- You need to implement simple maintenance procedures because you do not have a database administrator.
- You can risk losing modifications made to the data after the last differential backup, and it is more efficient to possibly have to recreate the data rather than use transaction log backups.

Note: During a full database or differential backup, enough of the transaction log will be backed up to produce a consistent database for when the database is restored.

Transaction Log Backups

A transaction log can be backed up separately from the database. It is important to back up the transaction log frequently. Doing this will ensure that you keep the most up-to-date backups inbetween performing complete database backups. A transaction log backup, like a database backup, can take place while the database is active.

Typically, transaction logs are backed up more often than databases. A transaction log might be backed up once a day, and a database might be backed up once a week. Backing up a transaction log takes less time and uses less storage space than backing up a database. (Do not back up a transaction log unless the database has already been backed up at least once.)

It takes longer to recover a database when you restore the database and all transaction logs sequentially than it does when you restore only the database. Finding the correct strategy depends on your environment. The time required to perform backups must be considered in relation to the time required to restore.

When a backup of a transaction log is performed, it is backed up from the last successfully executed log backup to the current end of the log. By default, the log will be truncated to the beginning of the active portion of the log, which contains the oldest open transaction. Truncating a transaction log removes the inactive portion of a log. You usually truncate a transaction log after you have backed up a database and a transaction log together.

Files/Filegroup Backups

You would choose to back up a file or file group when the database size and performance requirements make it impractical to perform a full database backup.

If you choose to back up a file instead of the full database, set procedures in place to assure that all files in the database are backed up regularly. Also, separate transaction log backups must be performed. After restoring a file backup, apply the transaction log, this rolls the contents of the file forward to make it consistent with the rest of the database. For further information, refer to the Microsoft SQL Server documentation.

The BACKUP statement requires that entire filegroups affected by a CREATE INDEX statement since the last filegroup be backed up as a unit.

- If an index is created on a filegroup, that entire filegroup must be backed up in a single backup operation. Backups of individual files that are part of the affected filegroup are not allowed.
- If an index is created on a filegroup different from the one that the table resides in, then both filegroups (the filegroup containing the table and the filegroup containing the newly created index) must be backed up.
- If more than one index is created on a filegroup different from the filegroup in which the table resides, all filegroups must immediately be backed up to accommodate these different filegroups.

The BACKUP statement detects all of these filegroup situations and communicates to the backup user the minimum filegroups that must be backed up.

Backup Options

The Backup Agent for Microsoft SQL Server allows the following types of backups to be performed:

- Complete Database— Performs a backup of your entire database.
- Differential— Performs a backup of data that has changed since the last complete backup was performed. (For example, if you ran a complete backup of your database on Sunday night, you might want to run a differential backup on Monday night to back up just those files that have changed on Monday.)
- Transaction Log— Performs a backup of your Transaction Log.
- Files and/or Filegroups— Performs a backup of selected files within a database. You should choose to back up a file or file group when the database size and performance requirements make it impractical to perform a full database backup.
- Database Consistency Check (DBCC)— Checks the physical and logical consistency of a database. The following 2 database consistency checks are performed when this backup option is enabled:

  DBCC CHECKDB**—** checks the allocation and structural integrity of all the objects in the specified database. By default it will perform a check for indexes which can increase the overall execution time. But the system table indexes will be checked anyway, with or without using this option.

  DBCC CHECKCATALOG**—** checks for consistency in and between system tables in the specified database.

  The Backup Agent provides the following four options with DBCC to be executed during database backup:

  - Before Backup—perform DBCC before backup of the database.
  - After Backup—perform DBCC after backup of the database.
  - Continue with backup, if DBCC fails—proceed with the database backup even if DBCC fails.
  - Do not check indexes—perform DBCC for the database without checking indexes for user defined tables.

  All of the error messages that occur during the DBCC will be displayed in the log file of the backup agent (dbasql70.log).

Backing Up

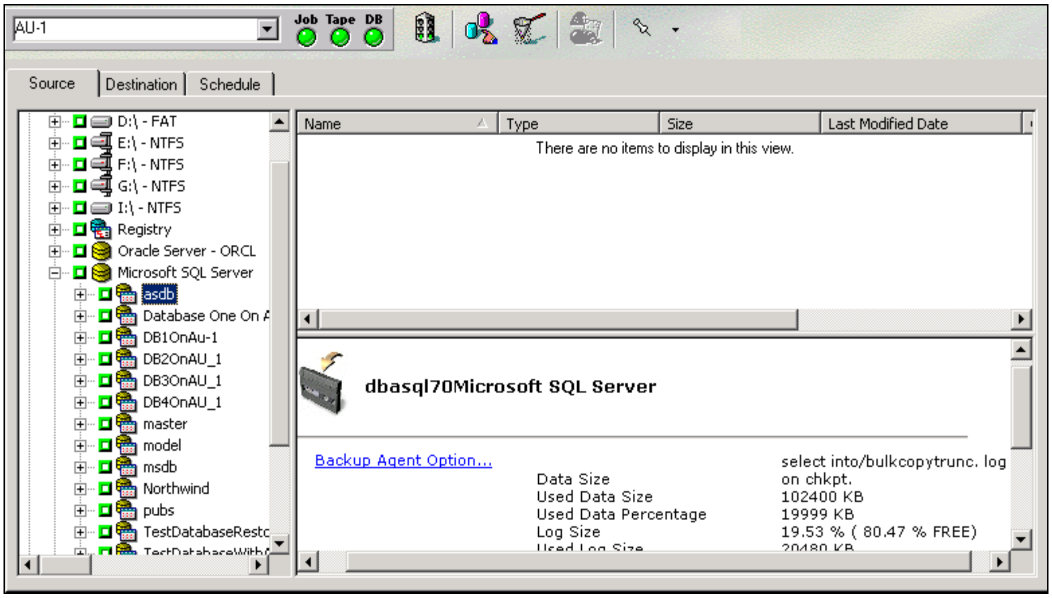
This section provides the basic steps you need to backup a database.

1. Make sure that the SQL Server is running on your server. The Microsoft SQL Server service must be started.

2. Start ARCserve and open the Backup Manager. Refer to the online *ARCserve Administrator Guide* for details on starting ARCserve. The Backup Agent is started automatically when you launch ARCserve.

3. From the browser in the Backup Manager, locate the Microsoft SQL Server. It is listed under the server on which it is installed.

4. Double-click on the SQL Server to display a list of databases for this server and then select a database.



There are no items to display in this view.

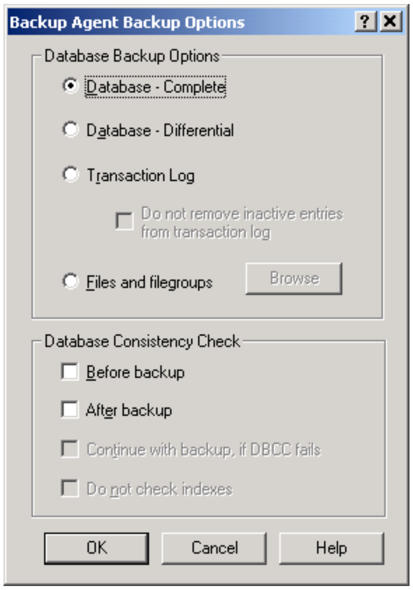
dbasql70Microsoft SQL Server

```
                    select into/bulkcopytrunc. log
Data Size           on chkpt.
Used Data Size      102400 KB
Used Data Percentage 19999 KB
Log Size            19.53 % ( 80.47 % FREE)
Used Log Size       20480 KB
```

Information about the selected database is displayed on the right side of the screen.

5. Right-click the database object in the browser. A pop-up menu opens.

6. Select Backup Agent Option. The Backup Agent Backup Options dialog opens:



Backup Agent Backup Options

Database Backup Options

● Database - Complete

○ Database - Differential

○ Transaction Log

□ Do not remove inactive entries from transaction log

○ Files and filegroups

Database Consistency Check

□ Before backup

□ After backup

□ Continue with backup, if DBCC fails

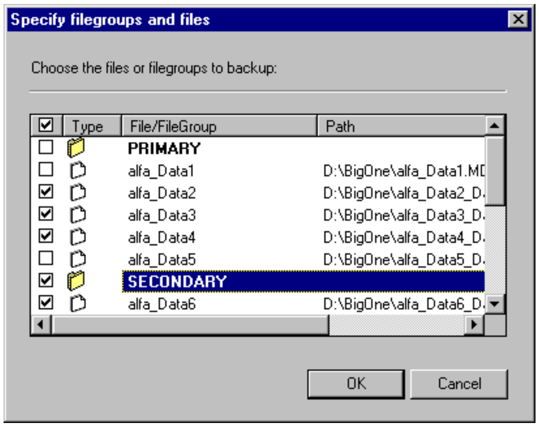
□ Do not check indexes

ARCserve provides the following choices when you back up a transaction log:

- Truncate the transaction log files after backup— The log files by default will be truncated.
- Do not remove inactive entries from transaction log— This choice keeps inactive log entries after backup.

7. Make your selections. Refer to the previous section on Backup Option for further information.

- Database - Complete— Performs a backup of your entire database.
- Database - Differential— Records only those data changes made to the database after the last full database backup.
- Transaction Log— If you choose to back up the transaction log, by default, it will truncate your inactive log entries. If you do not wish to truncate, check the "Do not remove inactive entries from transaction log" option.
- Files and Filegroups— Click on the Browse button to choose which files/filegroups you wish to back up. Click on the files or filegroups you wish to backup (a checkmark should appear).

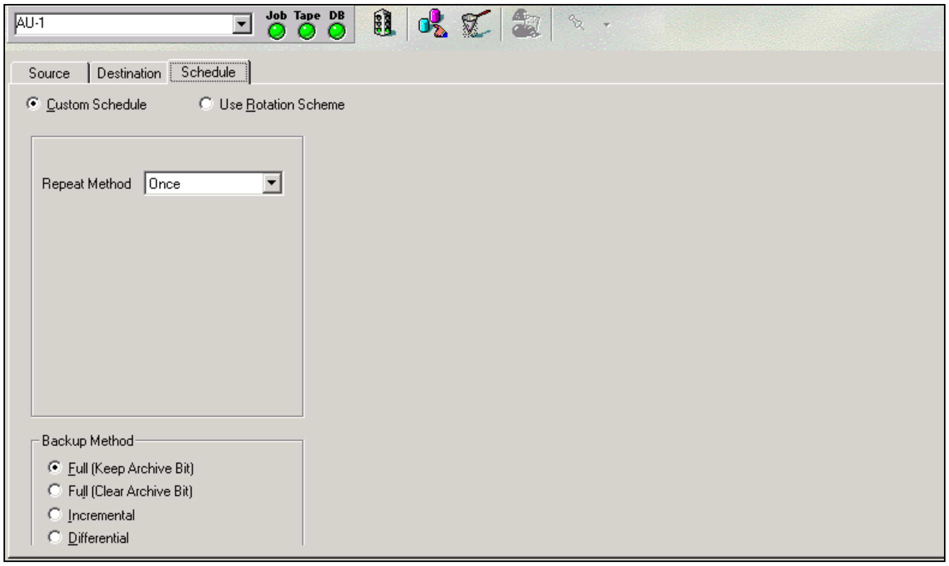


Specify filegroups and files

Choose the files or filegroups to backup:

☑	Type	File/FileGroup	Path
☐		PRIMARY	
☐		alfa_Data1	D:\BigOne\alfa_Data1.MI
☑		alfa_Data2	D:\BigOne\alfa_Data2_D.
☑		alfa_Data3	D:\BigOne\alfa_Data3_D.
☑		alfa_Data4	D:\BigOne\alfa_Data4_D.
☐		alfa_Data5	D:\BigOne\alfa_Data5_D.
☑		SECONDARY	
☑		alfa_Data6	D:\BigOne\alfa_Data6_D.

8. Click OK. You are now ready to submit a backup job.
9. Select a Backup Destination.
10. Click the Schedule tab. The Schedule dialog appears:



AU-1

● Custom Schedule ○ Use Rotation Scheme

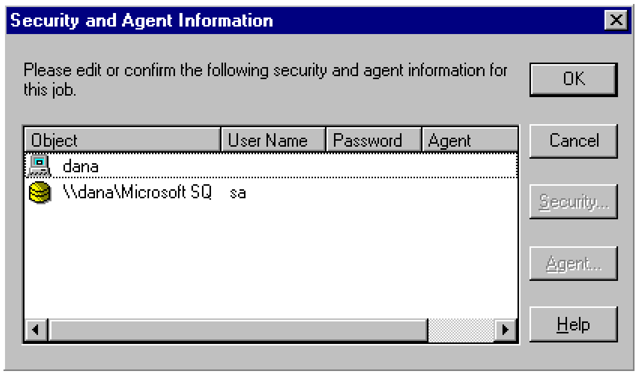
Repeat Method: Once

Backup Method

● Full (Keep Archive Bit)
○ Full (Clear Archive Bit)
○ Incremental
○ Differential

From this dialog, you can either run the job now or schedule the job for another time. Refer to the online *ARCserve Administrator Guide* for information on scheduling backups.

11. Click the Run button on the toolbar. The Security and Agent Information dialog box opens:



Security and Agent Information

Please edit or confirm the following security and agent information for this job.

Object	User Name	Password	Agent
dana			
\\dana\Microsoft SQ	sa		

Note: In this dialog box, the column and button labeled "Agent" refer to the Client Agent, not the Backup Agent. If you wish, you may enter Client Agent information at this time. Refer to the online *ARCserve Administrator Guide* for more information on Client Agents.

12. Enter the user name and password for the SQL Server. Click on Security to change the security information for the SQL Server. Enter the new user name and password in the Security dialog box, then click OK.

13. After verifying or changing the security information, click OK to continue. The Summit Job dialog box appears.

14. Click OK. The Job Status window opens. Use this window to monitor the current status of your job. Refer to the online *ARCserve Administrator Guide* for more information.

Restore Overview

To *restore* is to load a database from a backup of that database and (if applicable) one or more backups of its transaction log. If a database is lost or damaged, you can restore the database by reloading the most recent database and the successive log backups. A restore overwrites any information in the database with the backed-up information. Restores are accomplished by using ARCserve, the Backup Agent, and the SQL RESTORE statement.

When a database is restored, SQL Server rolls back any uncommitted transactions that were active at the moment the backup began. When the restore is complete, the database is in the same state it was in when the BACKUP statement was executed, minus any transactions that were active at that point.

Once the data from the backup has been reloaded, SQL Server reinitializes any remaining unused pages. For example, if a 100-MB database contains only 5 MB of data, all 100 MB of space is still rewritten, therefore a database restore command takes at least as long as the CREATE DATABASE statement that created the database.

A database locks so that it cannot be modified while the restore is in process. However, users can access and modify other databases on SQL Server during this time.

If a failure occurs while a database is being restored, SQL Server notifies the system administrator but does not recover the partially restored database. The database restore must be restarted.

The destination database must have at least as much storage space as was allocated to the backed up database. The actual amount of data in the backed up database is irrelevant. To find out about allocated storage space, use SQL Enterprise Manager or the DBCC CHECKALLOC statement.

After a media failure, restart SQL Server. If SQL Server cannot access a database, it marks the database as suspect, locks it, and displays a warning message. A damaged database must be dropped—a process that can be performed using the SQL Enterprise Manager.

During a restore operation, the selected database must not be in use. Any data in the selected database is replaced by the restored data.

The Backup Agent for Microsoft SQL supports the following:

- Restoring a complete database, which restores the entire database.
- Restoring a differential backup— Differential backups allow the database to be restored only to the point in time that the differential backup was created. It does not, however, restore to the exact point of failure as a transaction log restore does. Usually differential backups are supplemented by creating multiple transaction log backups after each database backup is created. By using a combination of database, differential backup, and transaction log backups, the time of database recovery and the amount of data loss due to failure can be minimized.
- Restoring a transaction log— Once you’ve restored a database, you can load the differential backup (if any), and transaction log backups that were accomplished after that database backup. Loading transaction logs allows you to recover as much of a database as possible, in case of a system failure. Backups of the transaction log must be loaded in the sequence in which they were created. SQL Server checks the timestamps on each backed up database and each backed up transaction log to see that the sequence is correct.

Restoring a transaction log is also referred to as “applying” a transaction log. Restoring a transaction log results in re-execution of the changes it contains and in rolling back any transactions that were uncommitted when the transaction log was backed up.

Once the entire sequence of transaction log backups has been loaded, the database is restored to its state at the time of the last transaction log backup, minus uncommitted transactions. (The exception to this is the SQL BACKUP LOG statement, which was active at the point of restore. It is not rolled back: the next RESTORE LOG statement completes it.)

- Restoring a file/filegroup— Files and filegroups can be restored from either a file/filegroup backup or from a full database backup operation. When restoring files/filegroups you must apply a transaction log. When a file/file group is restored, the transaction log must be applied to the database files immediately after the last file/filegroup operation.

  The “Force restore over existing database” option allows the replacement of files, but you will not be allowed to move the files to another location.

  The Recovery Completion State must be set to the Leave database nonoperational, but able to restore additional transaction logs option.

  This option is not supported by SQL Server for file/filegroup restores. As is mentioned in the SQL Server documentation, after restoring a file/filegroup, the user must apply a Log Session in order to roll the contents of the file forward to make it consistent with the rest of the database. One exception is if the files were not modified, the log does not need to be applied.

  However, this introduces a special case for ARCserve, when there are no Log Sessions to follow the File/Filegroup Session, ARCserve cannot verify whether the file was modified or not. Because of this it does not know which should be the final recovery completion state, so by default it chooses Leave database nonoperational, but able to restore additional transaction log. Therefore, every time you backup a file/filegroup, make sure you backup a log immediately afterwards.

  The RESTORE statement requires that one or more filegroups be restored in a single operation if indexes were created on these filegroups since the last time these files were backed up. This requirement of restoring the entire set of filegroup(s) is necessary whether the restore is from a filegroup backup or a full database backup. The RESTORE statement detects this filegroup situation and reports the minimum filegroups that must be restored.

For further information, refer to the Microsoft SQL Server documentation.

Differential Backup

If a differential backup is damaged or lost, you can restore the differential backup by reloading the most recent database backup and the succeeding last differential backups. Or if you have some corrupt data on the differential backup, you can reload a differential backup over the old differential backup.

Once a differential backup session has been selected to be restored, ARCserve's Automatic Selection option will *automatically select the appropriate Differential backup and Database Sessions, along with the proper options (if any)*. Automatic Selection will ensure that you have the correct Session(s) being restored in your job. Although you may package the appropriate Session(s) by yourself, Automatic Selection will save time.

When a differential backup is restored, the backup must not be in use. Any data in the specified differential backup is replaced by the restored data.

Unlike transaction log backups, differential backups do not allow a database to be restored to the exact point of failure--only to the point in time that the differential backup was created.

Transaction Log

When restoring the transaction log, you must remember to apply it to the appropriate database, differential backup, or filegroup Session. In summary, do the following:

1. Restore the Database Session.
2. Restore the latest Differential backup Sessions, if any.
3. Restore the transaction log.

Once a transaction log has been selected to be restored, however, the Automatic Selection option will automatically select the appropriate Transaction Log, Differential backup and Database Sessions, along with the proper options, if any. Automatic Selection will ensure that you have the correct sessions being restored in your job. Although you may package the appropriate sessions by yourself, Automatic Selection can ultimately save time.

Files and Filegroups

Individual files and/or filegroups can be restored from either a file/filegroup backup or from a full database backup operation. When restoring a file/filegroup, you must remember to apply it to the appropriate transaction log.

Note: When this option is used, the transaction log must be applied to the database immediately after the last file/filegroup Restore operation. This allows the contents of the file to roll forward making it consistent with the rest of the database.

When choosing which file/filegroup to restore, perform the following steps:

- Restore the file/filegroup.
- Apply the appropriate transaction log sessions.

Once a file/filegroup has been selected to be restored, however, the ARCserve's Automatic Selection option will automatically select the appropriate transaction log, along with the proper options, if any. Automatic Selection will ensure that you have the correct sessions being restored in your job. Although you may package the appropriate sessions by yourself, Automatic Selection can save time.

Restoring a Master Database

Restoring the master database is a special case. You will need to rebuild the master database from the SQL server setup. (Refer to your Microsoft SQL Server documentation for complete instructions). Start SQL server in single-user mode, then restore the master from the most recent backup.

To restore the master database:

1. Start SQL Server in Single User Mode, using the –m option.
2. Stop the SQL Server Agent Service.
3. Submit a restore job of the master database backup.

For further information on this type of restore, as well as other special cases, please refer to *SQL Server Books Online*.

You can also back up a database and restore that backup into another database.

Important! If the msdb database is to be restored from a backup, the SQLagent Service must be stopped before the restore begins and then restarted after the restore is complete. If this is not done, then SQLagent Service will not function properly.

Restore Options

The Backup Agent for Microsoft SQL Server allows the following types of restores to be performed:

ARCserve Automatic Selection

This option performs automatically, selecting dependent sessions and options, saving time and possible error in packaging the restore job.

Automatic Selection also chooses the appropriate options for the desired restore job so you can restore an entire database and its logs successfully.

Restore Type

This option allows you to select the desired type of restore:

- The Database option is used for restoring complete databases, differential backups, and transaction log backups.
- The Files or filegroups option is used for restoring files or filegroups backups or files belonging to database backups, but not for log backups or differential backups. Once selected, you will be prompted to choose which files you want to restore.
- Force Restore Over Existing Database— Force restore, which is applied only to databases and files, instructs the SQL Server to replace the existing database with a new one it creates. Force Restore is also available when restoring Transaction Log sessions if Automatic Selection is checked.
- Partial Restore— Specifies a partial restore operation. SQL Server provides a mechanism to restore part of the database to another location so that a damaged or missing data can be copied back to the original database. The granularity of the partial restore operation is the database filegroup. The primary file and filegroup are always restored, along with the files that you specify and their corresponding filegroups. The result is a subset of the database. Filegroups that are not restored are marked as offline and are not accessible.

  Note: This option is for SQL Server 2000 only.

Log Point in Time Restore

This option specifies that the database will be restored to the state it was in as of the specified date and time. Only transaction log records written before the specified date and time are applied to the restored database. The Automatic Selection feature is recommended for this option.

In order to find the correct log, SQL Server restores the record of each transaction log that contains the start and finish time of the backup. Meanwhile, SQL Server will search the record for the time selected. If SQL Server does not find the correct time, the log will not be recovered. If SQL Server does find the correct time, however, it will be restored to the point in the record that contains the time submitted by the user. Once this is done, the Agent will signal ARCserve to stop restoring. The database will be then be fully recovered.

If there are other logs with the same time, those logs are ignored, the subsequent sessions are skipped, and the restore jobs will be declared "Incomplete".

The Log point in time restore option does, however, have its limitations. For example, with Automatic Selection checked, you choose the "Force restore over database" option again and select a single log or a couple of logs belonging to the same database to be restored but fail to select the appropriate database, differential backup, filegroup sessions to be restored first. In this case the job will be "Incomplete", and the subsequent sessions for that database will be ignored.

The options available for Log point in time restore running on SQL 2000 are:

- Stop before log mark— This option includes an After datetime field where the user can specify a specific date and time. The option specifies recovery to the specified mark but **does not include the transaction that contains the mark**. If After datetime is omitted, recovery stops at the first mark with the specified name. If After datetime is specified, recovery stops at the first mark having the specified name exactly at or after datetime.

  Note: This option is for SQL Server 2000 only.

- Stop at log mark— This option includes an After datetime field where the user can specify a specific date and time. The option specifies recovery of the database to the specified mark, **including the transaction that contains the mark**. If the After datetime field is omitted, recovery stops at the first mark with the specified name. If the After datetime is specified, recovery stops at the first mark having the specified name exactly at or after datetime.

  Note: This option is for SQL Server 2000 only.

- Stop at time— This option includes a date and time field where the user can specify a specific date and time. The option specifies recovery of the database to the specified date or time. This is the default option for Log point in time restore.

Restore Database Files as

This option has three purposes:

- Show the backed up files and the appropriate information.

- Allow a user to select the files to be restored in the case of a File/Filegroup Backup Session or a Database Backup Session. This can be done by selecting the "File or filegroups" Restore type.

- Allow a user to change the path of these files and to move them to another location by editing the "Move File to Physical Location" column.

Note: For SQL Server 2000 users, you may also select log files and change the path and move to another location.

Note: Remember that the move function is only for a complete database restore.

Recovery Completion State

This option allows you to specify the final state of a session restore via the following selections:

- Leave database operational. No additional transaction logs can be restored— This option instructs the restore operation to rollback any uncommitted transactions. After the recovery process, the database is ready for use.

  Note: If you choose to use Automatic Selection, no manual selection for the subsequent sessions for these options is required because ARCserve will perform the selection of sessions and the necessary options automatically. If not, you must follow the SQL Server rules regarding the restore flow. For further information, refer to the SQL Server documentation.

- Leave database nonoperational, but able to restore additional transaction logs— When this option is selected, it instructs the restore operation to not roll back any uncommitted transactions. Either this option or option C must be specified if another transaction log has to be applied. SQL Server requires that this option be used on all but the final statement. It's also required on a database backup and multiple transaction log backups, or when multiple statements are needed (for example, a full database backup followed by a differential backup). Again, you must take the same note as above into consideration.
- Leave database read-only and able to restore additional transaction logs— This option means that a standby (warm backup) server is a second server that can be brought online in the event of failure of the primary production server, which will contain a copy of the databases on the primary server. For further information about standby servers refer to the SQL Server documentation.
- Database Consistency Check (DBCC)— Checks the physical and logical consistency of a database. The following two database consistency checks are performed when this option is enabled:

  DBCC CHECKDB—checks the allocation and structural integrity of all the objects in the specified database; by default it will perform a check for indexes which can increase the overall execution time. But the system table indexes will be checked anyway with or without using this option.

  DBCC CHECKCATALOG—checks for consistency in and between system tables in the specified database.

The Backup Agent provides the following two options with DBCC to be executed during database restore:

- After Restore—perform DBCC after restore of the database.

- Do not check indexes—will perform DBCC for the database without checking indexes for user defined tables.

Restricted User Access After Restore

Note: This option is for SQL Server 2000 only.

Restricts access for the newly restored database to members of the db_owner, dbcreator, or sysadmin roles. In SQL Server 2000, RESTRICTED_USER replaces the DBO_ONLY option from SQL7. Use with the **Leave database operational, No additional transaction logs can be restored** option.

Keep Replication Settings

Note: This option is for SQL Server 2000 only.

Instructs the restore operation to preserve replication settings when restoring a published database to a server other than that on which it was created. Keep replication settings is to be used when setting up replication to work with log shipping. It prevents replication settings from being removed when a database or log backup is restored on a warm standby server and the database is recovered. Specifying this option when restoring a backup with the **Leave database nonoperational, but able to restore additional transaction logs** option is not permitted. This option should be used only with the **Leave database operational, No additional transaction logs can be restored** option.

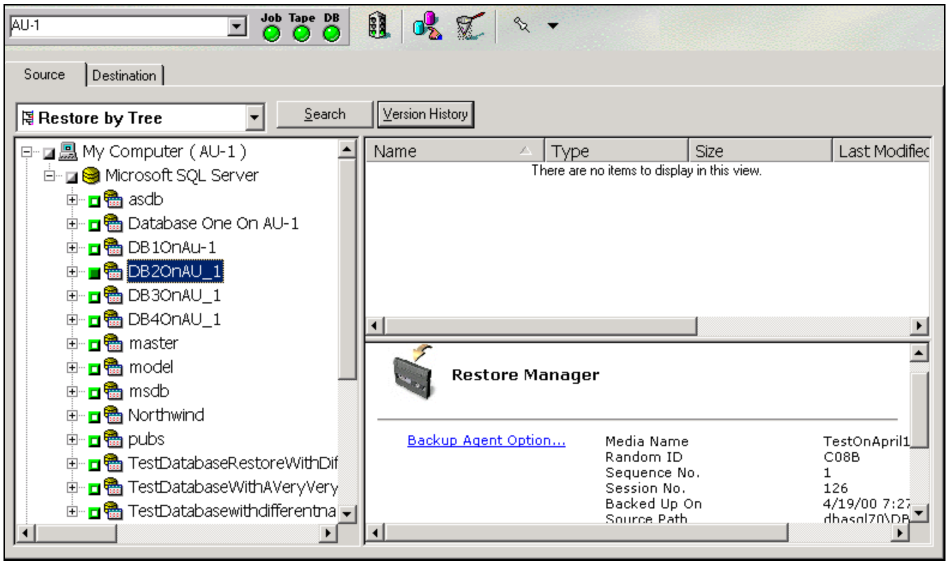
Restoring

If a database is damaged or lost, you can restore the database by reloading the most recent database backup, the latest differential backup, and the succeeding transaction log backups. Or, if you have some corrupt data on the database, you can reload a database backup over the old database.

When a database is restored from a backup, that database must not be in use because any data in the specified database is replaced by the restored data.

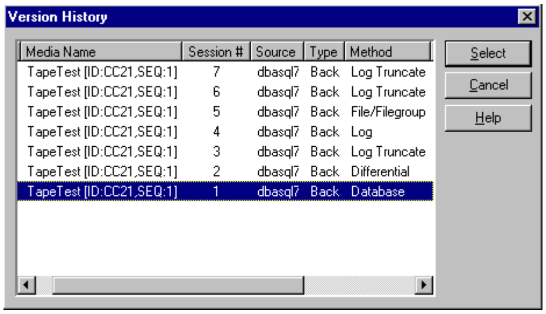
To Restore a Complete Database, Differential, Transaction Log or File/Filegroup Backup

1. Start ARCserve and open the Restore Manager. Refer to the online *ARCserve Administrator Guide* for details on starting ARCserve.
2. From the Source tab, select the Restore by Tree option.
3. In the browser, highlight the Database Name you want to restore.



Media Name	TestOnApril1
Random ID	C08B
Sequence No.	1
Session No.	126
Backed Up On	4/19/00 7:27
Source Path	dbasql70\DB

4. Click the Version History button. The following dialog box opens:



Media Name	Session #	Source	Type	Method
TapeTest [ID:CC21,SEQ:1]	7	dbasql7	Back	Log Truncate
TapeTest [ID:CC21,SEQ:1]	6	dbasql7	Back	Log Truncate
TapeTest [ID:CC21,SEQ:1]	5	dbasql7	Back	File/Filegroup
TapeTest [ID:CC21,SEQ:1]	4	dbasql7	Back	Log
TapeTest [ID:CC21,SEQ:1]	3	dbasql7	Back	Log Truncate
TapeTest [ID:CC21,SEQ:1]	2	dbasql7	Back	Differential
TapeTest [ID:CC21,SEQ:1]	1	dbasql7	Back	Database

Version History gives information on the media name attached to the selected source, its backup Session number, backup method, as well as the date and time the backup was made. Using Version History allows you to select the session to restore from on the backup media.

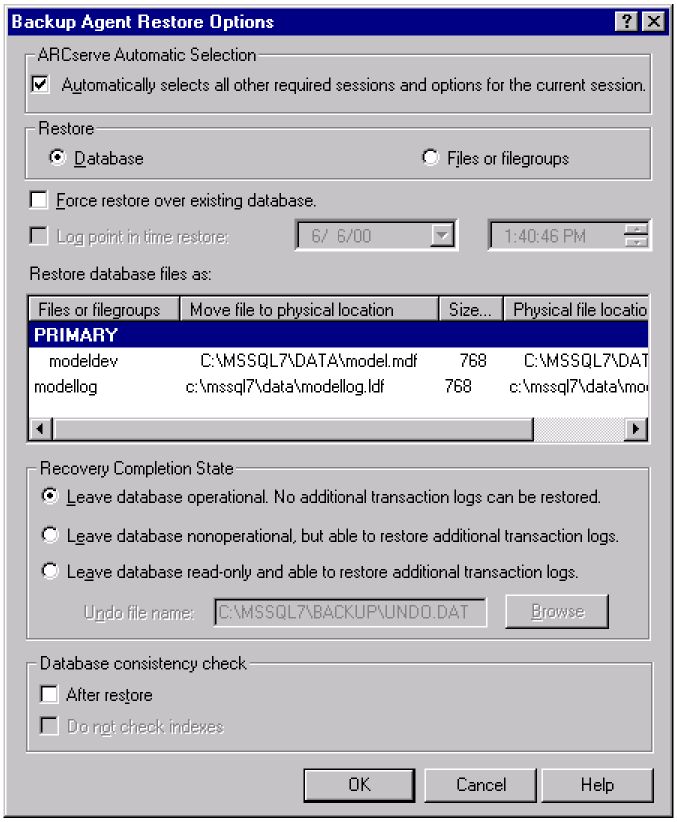
5. Select a backup session for this database to be restored and click the Select button. The Version History dialog box closes.

6. In the browser, select the Database Name you highlighted in Step 3. The box to the left of the source will fill green, indicating it has been selected.

7. Right-click the Database Name in the browser. A pop-up menu opens.

8. Select Backup Agent Option. The Backup Agent Restore Options dialog opens:

Screen will vary depending on the Restore session(s) selected.



Backup Agent Restore Options

ARCserve Automatic Selection

☑ Automatically selects all other required sessions and options for the current session.

Restore

● Database ○ Files or filegroups

□ Force restore over existing database.

□ Log point in time restore: 6/ 6/00 1:40:46 PM

Restore database files as:

Files or filegroups	Move file to physical location	Size...	Physical file locatio
PRIMARY			
modeldev	C:\MSSQL7\DATA\model.mdf	768	C:\MSSQL7\DAT
modellog	c:\mssql7\data\modellog.ldf	768	c:\mssql7\data\mo

Recovery Completion State

● Leave database operational. No additional transaction logs can be restored.

○ Leave database nonoperational, but able to restore additional transaction logs.

○ Leave database read-only and able to restore additional transaction logs.

Undo file name: C:\MSSQL7\BACKUP\UNDO.DAT

Database consistency check

□ After restore

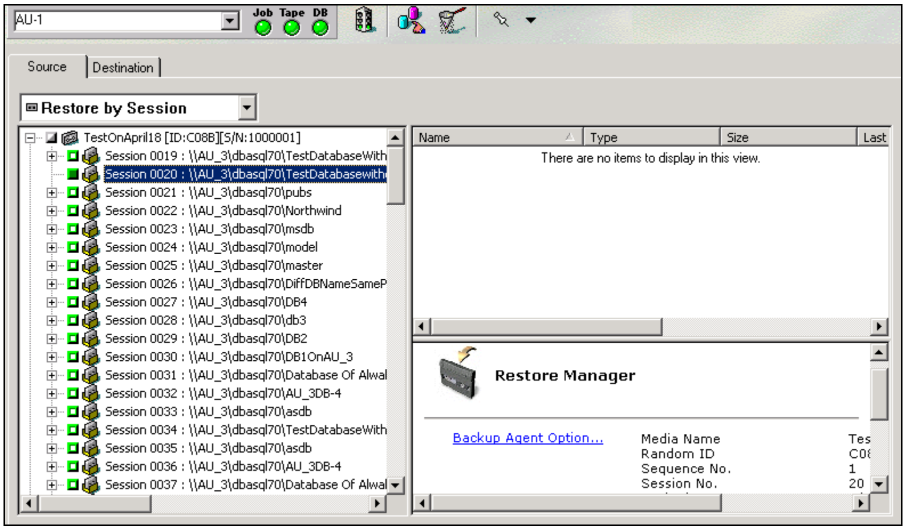
□ Do not check indexes

Note: The screen may vary depending on the Restore sessions selected.

Tip: The Automatic Selection option is chosen by default. This chooses the appropriate options for you automatically (the options chosen will depend on the type of restore). To accept the default, click OK. To review and change your options, refer to the “Restore Options” section later in this chapter.

Restore by Session Method Only

In the browser, select the Session you want to restore.



Restore by Session

TestOnApril18 [ID:C08B][S/N:1000001]
Session 0019 : \\AU_3\dbasql70\TestDatabaseWith
Session 0020 : \\AU_3\dbasql70\TestDatabasewith
Session 0021 : \\AU_3\dbasql70\pubs
Session 0022 : \\AU_3\dbasql70\Northwind
Session 0023 : \\AU_3\dbasql70\msdb
Session 0024 : \\AU_3\dbasql70\model
Session 0025 : \\AU_3\dbasql70\master
Session 0026 : \\AU_3\dbasql70\DiffDBNameSameP
Session 0027 : \\AU_3\dbasql70\DB4
Session 0028 : \\AU_3\dbasql70\db3
Session 0029 : \\AU_3\dbasql70\DB2
Session 0030 : \\AU_3\dbasql70\DB1OnAU_3
Session 0031 : \\AU_3\dbasql70\Database Of Alwal
Session 0032 : \\AU_3\dbasql70\AU_3DB-4
Session 0033 : \\AU_3\dbasql70\asdb
Session 0034 : \\AU_3\dbasql70\TestDatabaseWith
Session 0035 : \\AU_3\dbasql70\asdb
Session 0036 : \\AU_3\dbasql70\AU_3DB-4
Session 0037 : \\AU_3\dbasql70\Database Of Alwal

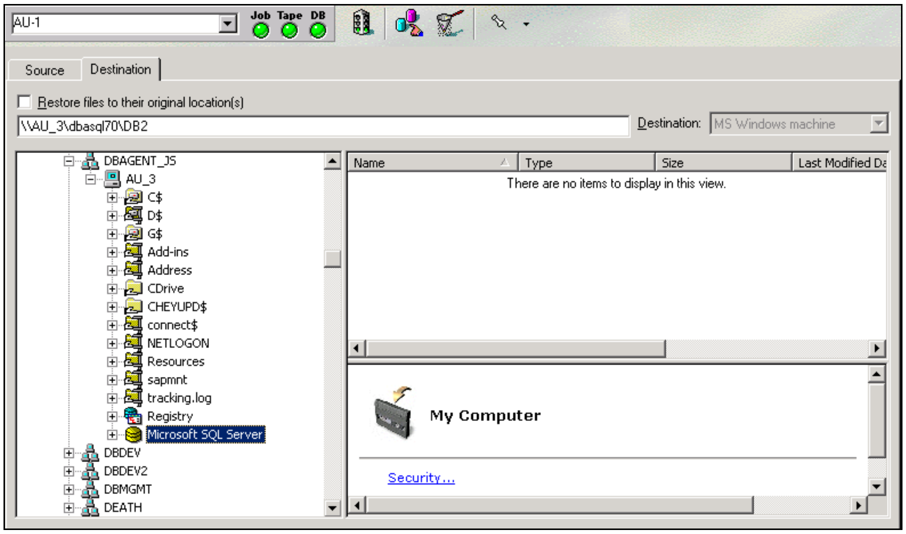
There are no items to display in this view.

9. Click the Destination tab and in the browser, select a destination.

You can select to restore either to the original server that was backed up or to a different server. In either case, you can restore the database with the same name as the original or with a different name.

To Restore to a Different Server Using the Same Database Name

Clear the "Restore files to their original location(s)" check box.



□ Restore files to their original location(s)

\\AU_3\dbasql70\DB2

Destination: MS Windows machine

There are no items to display in this view.

To Restore to a Different Server Using a Different Database Name

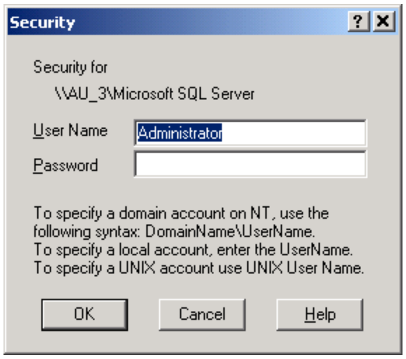
Clear the "Restore files to their original location(s)" check box. Select the server you want to be the destination. Select the Microsoft SQL Server icon. In the space below the check box, enter the new name of the database at the end of the path displayed.

10. Click the Run/Schedule button on the toolbar.

12. Specify security information according to the type of Restore operation that you selected.

To Specify Security When Restoring to the Original Server

Enter the user name and password for your SQL Server. Click OK.



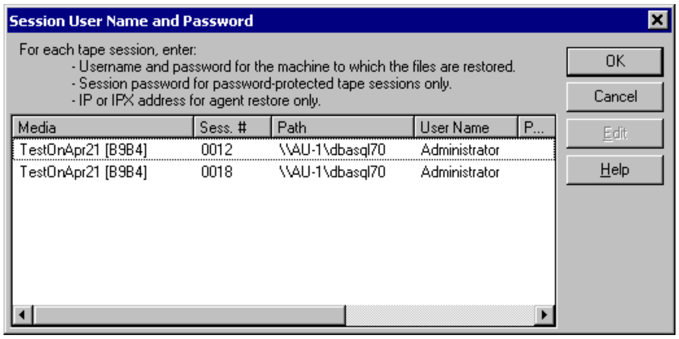
Security for
\\AU_3\Microsoft SQL Server

User Name: Administrator

Password:

To specify a domain account on NT, use the following syntax: DomainName\UserName.
To specify a local account, enter the UserName.
To specify a UNIX account use UNIX User Name.

The Session user name and password dialog box opens. Verify or change the user name and password for the Windows NT/2000 machine on which the SQL Server is loaded. To change the user name or password, select the Session and click Edit. If a Session password was assigned to this Session, enter the Session password.

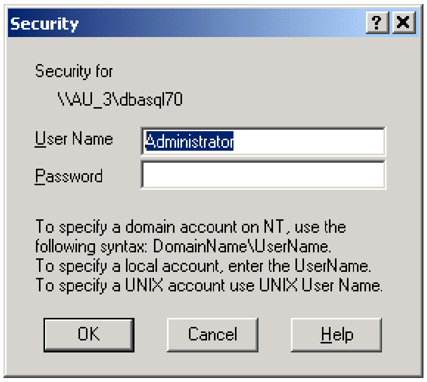


For each tape session, enter:
- Username and password for the machine to which the files are restored.
- Session password for password-protected tape sessions only.
- IP or IPX address for agent restore only.

Media	Sess. #	Path	User Name	P...
TestOnApr21 [B9B4]	0012	\\AU-1\dbasql70	Administrator	
TestOnApr21 [B9B4]	0018	\\AU-1\dbasql70	Administrator	

To Specify Security When Restoring to a Different Server

Enter the user name and password for the Windows NT/2000 server. Click OK.


Security

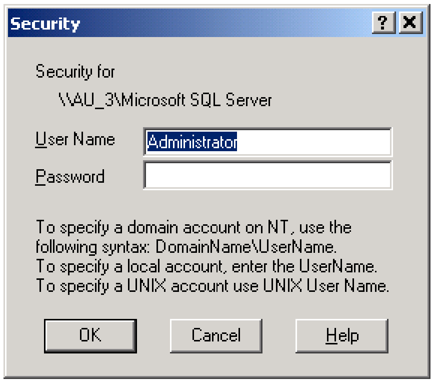
Security for
\\AU_3\dbasql70

User Name: Administrator

Password:

To specify a domain account on NT, use the following syntax: DomainName\UserName.
To specify a local account, enter the UserName.
To specify a UNIX account use UNIX User Name.

The Security dialog box for the SQL Server appears. Enter the user name and password for your SQL Server.


Security

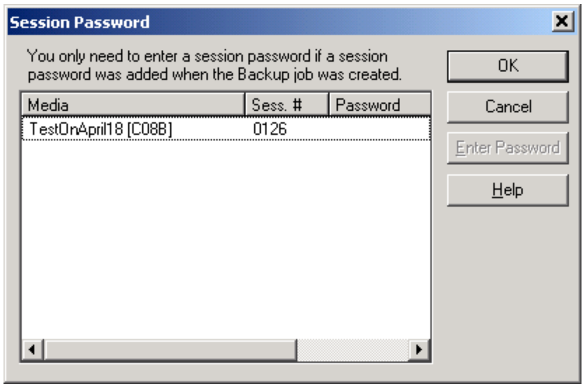
Security for
\\AU_3\Microsoft SQL Server

User Name: Administrator

Password:

To specify a domain account on NT, use the following syntax: DomainName\UserName.
To specify a local account, enter the UserName.
To specify a UNIX account use UNIX User Name.

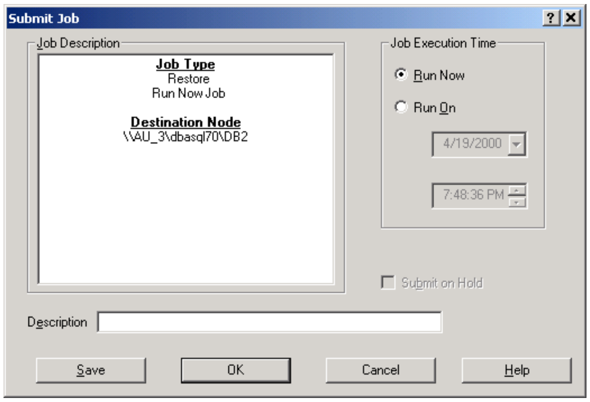
Click OK. The Session dialog box opens. If a Session password was assigned to this Session, enter the Session password. Click OK.


Session Password

You only need to enter a session password if a session password was added when the Backup job was created.

Media	Sess. #	Password
TestOnApril18 [C08B]	0126	

14. Click OK. The Submit Job dialog appears.



Job Type
Restore
Run Now Job

Destination Node
\\AU_3\dbasql70\DB2

● Run Now
○ Run On
4/19/2000
7:48:36 PM
□ Submit on Hold

15. Click OK to submit this job. The Job Status window opens. Use this window to monitor your job.

Backing Up and Restoring in a SQL Server Cluster Environment

ARCserve supports the Active/Passive configuration of the SQL Virtual Server in the Microsoft Cluster Server (MSCS) environment.

A virtual SQL Virtual Server is backed up and restored just like any non-clustered SQL Server using ARCserve's Backup Agent with the following important differences:

- ARCserve's Backup Agent must be set up on the local drives of all nodes in the SQL Virtual Server.
- The SQL Virtual Server is always remote. The SQL Virtual Server is listed under the MSCS icon. Therefore it must be backed up through the network or the preferred shares under the MSCS icon in the Backup Manager. It should not be backed up through the local machine or the SQL Virtual Server icon.
- If the node that the SQL Virtual Server is currently running in fail over during a backup job, the backup job will fail and must be resubmitted.

Appendix

A Troubleshooting and Disaster Recovery

Error Messages

This section explains the most common error messages for ARCserve and the Backup Agent:

Error Description	Cause/Solution
Backup or Restore Failed	**Probable Cause:** There may be a number of reasons for backup or restore to fail. **Solution:** Please refer to the Valid Operations with Database Options' section below to determine when a backup or a restore is allowed. Check the Backup Agent log file called DBASQL60.LOG, or DBASQL70.LOG which is located in the Backup Agent directory. Refer also to your SQL Server manual for information on backup operations.

No Microsoft SQL Server icon in ARCserve browser	**Probable Cause:**
	■ The Backup Agent RPC Server is not running or is dead.
	■ The Backup Agent is not installed.
	■ No Backup Agent entry exists in the registry. It should be found at:
	`SOFTWARE\ComputerAssociates\ARCserve\DSAgent\`
	`CurrentVersion\agent\dbasql60`
	Or
	`SOFTWARE\ComputerAssociates\ARCserve\DSAgent\`
	`CurrentVersion\agent\dbasql70`
	Solution:
	■ Restart the Backup Agent RPC Server.
	■ Install the Backup Agent.
	■ Re-install the Backup Agent.

Error Description	Cause/Solution
The Backup Agent appears to hang when double clicking on it. Backup Agent Error - (82)	**Probable Cause:** SQL Server is not running. **Solution:** Start SQL Server.
Remote pipe open failed: time-out. Backup Agent Error - (83)	**Probable Cause:** The Backup Agent RPC Server on the remote machine has a problem. **Solution:** Restart the Backup Agent RPC server on the remote machine. If it continues to fail, consider rebooting the remote machine.
Backup Agent Error - (85)	**Probable Cause:** Incorrect password submitted. **Solution:** Re-enter password for the database server.
E8602 Failed to read from database. Backup Agent Error - (73)	**Probable Cause:** ■ trunc. log on chkpt is enabled and attempting to backup a log ■ select into/bulkcopy is set and attempting to backup a log **Solution:** Disable the database option (for both cases).
E8604 Failed to start backup - Backup Agent Error - (73)	**Probable Cause:** A database has been marked as read-only and an attempt to backup the transaction log was made. **Solution:** Disable the database option.

Restore of the master database failed	**Probable Cause:** Restoring the master database requires SQL Server to be started in single user mode.
	Solution: Start SQL Server from a command line. For SQL 6.5 users:
	`e:\sq60\binn>sqlserver /c /de:\sql60\data\master.dat /m`
	For SQL 7.0 users:
	`e:\sq70\binn>sqlserver -m`
	Check your SQL Server Database Administrator guide for further assistance.

Backup Agent Messages

Backup Agent Message	Explanation
Backup agent Message (146)	The required time was not found in this log session or the database is in an unrecovered state.
Backup agent Message (143)	The current log session contains the time required by the user. The database was fully recovered, so this session will be the last log session restored for the current job.
Backup agent Message (144)	The current log session will be skipped due to a STOPAT restore option found in a previous session.

Valid Operations with Database Options

The following table indicates when a backup or restore can be performed if the various database options are set:

	BACKUP		RESTORE	
Option	Database	Log	Database	Log
Read only	YES	NO (SQL 6.5) YES (SQL 7.0)	YES	YES
Dbo use only	YES	YES	YES	YES
No chkpt on recovery	YES	YES	YES	YES
Don't recover	NO	NO	NO	NO
Not recovered	NO	NO	NO	NO
trunc.log on chkpt.	YES	NO	N/A	N/A
Single user	YES*	YES*	YES	YES
Select into/bulkcopy	YES	NO	N/A	N/A

* Provided that the single connection to the database is not in use.

Replication

According to Microsoft, the replication capability of SQL Server is not specifically designed to accomplish hot backups. Please be sure to read the SQL Server Database Administrator's guide on how to backup and, more importantly, restore in a replication scenario.

SQL Server Disaster Recovery

The Backup Agent for SQL Server utilizes SQL Server's Backup and Restore database functions. They do not actually back up the physical files that make up the database. This means that, in order to do a restore, the database must exist and the load command will restore the data into it.

In the case where a database has been dropped and you wish to restore it, there are two options:

- You can manually recreate the devices that make up the database and then create the database on them. Then it is a matter of restoring the database from tape, as described in this guide.

- The Backup Agent can automatically recreate the database. The limitation in this approach is that the Backup Agent only has knowledge of whether the database has a separate transaction log or not. So it will create a database on a single device only if the original database had data and log share devices. And it will create only one device large enough to hold all the data. Similarly, if the original database had the transaction log on separate devices, the agent will create a database with a database device and a log device large enough to hold all the data (independently of the total number of devices whose size summation originally made up the database).

Another point that must be taken under consideration is the master database. The master database should exist for SQL Server to run. With reference to the preceding information, it follows that:

- For SQL Server to run, a master database must exist.
- To have a master database the user must either reinstall SQL Server, rebuild the master database using SQL Server Setup, or restore an offline copy of it from tape.
- Once the master database exists, SQL Server MUST be running in order to execute the Restore command.

In addition, to restore the master database, SQL Server MUST be running in Single User mode. This can be done by starting the SQL Server Service as follows:

For SQL 6.5: `sqlserver /c /dcomplete_path_to_master_device /m`

For SQL 7.0: `sqlserver -m`

The master database keeps track of all resources allocated to SQL Server. For example, suppose we have a SQL Server configuration with five databases in addition to the master database. The master database is backed up. Then one database is dropped and the files that made it up are deleted. If the master database backup is restored, it will contain the information for the dropped database, which does not exist. Thus, SQL Server will mark it as "suspect" (that is, inaccessible by users). This database needs to be dropped again.

To avoid problems, we highly recommend that you perform at least one offline backup. We also recommend that every time there is a major change in the SQL Server configuration (database creation, device addition, or dropping a database) you also do an offline backup.

Note: The above suggestions are not requirements, but following them will speed up and simplify the disaster recovery procedure.

For further information, refer to the Microsoft SQL Server documentation.

Disaster Recovery in the Cluster Environment

- Set up the SQL server, but do not setup the Symmetric Virtual Server for the SQL server.
- Reinstall the Backup Agent.
- Start the SQL server in single user mode as described in the Disaster Recovery above.
- Restore the master database of the SQL server.
- Restart the SQL server to restore the rest of the databases.
- Set up the Symmetric Virtual Server for the SQL server.
- Reinstall the Backup Agent to enable the Symmetric Virtual Server related settings.

Disaster Recovery Scenario

Putting all the pieces together, the disaster recovery scenario looks like this:

1. Reinstall the Windows NT if necessary.
2. Reinstall ARCserve if necessary.
3. If an offline backup exists, restore it. Otherwise, reinstall SQL Server.
4. Reinstall the Backup Agent if necessary.
5. Complete the recovery according to which course of action you took in Step 3 above.

   If an offline backup was restored:

   - Start SQL Server in Single User mode and restore the master database. After the restoration is complete, SQL Server automatically stops.
   - Restart SQL Server as normal.
   - Restore all user databases and transaction logs. When this step is finished, recovery is complete.

   If SQL Server was reinstalled, then:

   - Manually recreate all devices and databases.
   - Restore the individual databases. When this step is finished, recovery is complete.

Note: If you are unable to carry out Steps 3 and 5, then use the Backup Agent to recreate dropped databases. To do this, you need to select each database, and submit the restore job. The Backup Agent will recreate the database and restore the data. If you select this method, the dropped database will always be recreated in the SQL Server\DATA subdirectory. When you have finished this step, recovery is complete.

ARCserve Limitations

Here are known limitations you should be aware of when using ARCserve and the Backup Agent for Microsoft SQL Server.

- ARCserve (like Windows NT) does not allow special characters to be used in file names, filegroup names, and database names.

  Examples include: /, *, <>, ?

  If you have named a SQL Database with one of these characters, by default, SQL Server will name the file/filegroup, or database with the same characters. To perform a backup or restore job, you must first rename the file/filegroup or database so it does not include one of these characters.

- When performing a "Log point in time restore", be aware of the STOPAT command which instructs ARCserve to stop restoring logs once the first one is found with the time desired by the user, even though there may be several detected with the time selected.

- When restoring a file and/or filegroup, if there are no Log Sessions to follow the File/Filegroup Session, ARCserve cannot verify if the file was modified or not. Because of this it does not know which should be the final recovery completion state, so by default it chooses Leave database nonoperational, but able to restore additional transaction log. Each time you back up a file/filegroup make sure you back up a log immediately afterwards.