

The ERASE command

Use the `ERASE` command with the SQL Backup Pro `-SQL` parameter to delete backup files based on their age, or the number of files you want to keep using the command line or extended stored procedure. The `ERASE` command works in a similar way to the `ERASEFILES` parameter used with the `BACKUP` and `RESTORE` commands, but enables you to delete backup files independently of a backup or restore operation.

You must also specify the database name, backup type, and a `PASSWORD` parameter if the files are encrypted.

Only files in the directory specified in the `FROM DISK` parameter are considered for deletion. Subdirectories are not checked.



When using the extended stored procedure the parameter or set of parameters (such as `-SQL`) must be delimited by single quotes. Therefore, wherever a single quote is used for the arguments below, for the extended stored procedure you must use **two** single quotes so that SQL Server does not interpret it as a string delimiter. See [Using the extended stored procedure](#) for more information.

Syntax

```
ERASE { FULL_BACKUPS | DIFF_BACKUPS | LOG_BACKUPS }  
FOR { database_name }  
FROM {DISK} = { 'physical_device_name' } [,...n]  
KEEP = { days | hours{h} | latest{b}}  
[ WITH  
  [[,]] PASSWORD = { 'password' | 'FILE:file_path' } ]  
]
```

Arguments

KEEP

You can choose to delete SQL Backup files based on:

- Age: files older than the specified number of days or hours are deleted. Type a number for days, or type *h* after the number for hours. For example, `KEEP = 24` deletes files that are more than 24 days old; `KEEP = 24h` deletes files that are more than 24 hours old. Note that a 'day' is calculated as a period of 24 hours, and takes no account of calendar date.
- Number of backups to keep: only the latest 'x' backups will be kept. To specify the number of backups to be kept, type *b* after the number. For example, `KEEP = 5b` ensures the latest 5 backups are kept; older backups are deleted.



If SQL Backup Pro cannot list the contents of the folder that contains the files to be deleted, it cannot delete the files. Ensure that the user account from which you are running SQL Backup Pro has permissions to list the folder contents.

PASSWORD

Specifies the password to be used with encrypted backup files. (Specifying a password does not prevent unencrypted files from being deleted.)

If the `ERASE` command applies to more than one backup file, and these have different passwords, then only those backup files for which the supplied password is correct will be deleted.



Storing your password in a plain text file (only available in SQL Backup 7.5 and later)

If you don't want the password to be stored in your SQL Agent jobs, you can use a password stored in a plain text file instead. This means access to the password can be restricted using Windows file permissions.

To do this, specify the file path and name after the `PASSWORD` keyword instead of the password itself.

Example

```
PASSWORD = 'FILE:C:\mypasswords\password.txt'
```

SQL Backup will read only the first line of text in the file (up to the first line return), and ignore everything after.

Example

This example keeps the 7 most recent full backups of the *pubs* database in the folder `C:\Backups`; older full backups of the *pubs* database are deleted from the folder:

```
SQLBackupC.exe -SQL "ERASE FULL_BACKUPS FOR [pubs] FROM DISK = 'C:\Backups' KEEP = 7b WITH PASSWORD = 'MyPassword'"
```

```
EXECUTE master..sqlbackup '-SQL "ERASE FULL_BACKUPS FOR [pubs] FROM DISK = ''C:\Backups'' KEEP = 7b WITH PASSWORD = ''MyPassword'''"
```