The CONVERT command

Use the `CONVERT` command with the SQL Backup Pro `-SQL` parameter to convert SQL Backup backup files (.sqb) to Microsoft Tape Format (MTF) files (.bak) using the command line or extended stored procedure. You can use the native SQL Server `RESTORE` command to restore MTF files.

Alternatively, you can use the SQL Backup File Converter (either from the command line, or using the GUI application). This application can also convert SQL Backup v7 or v6 files to SQL Backup v5 files; both file types have extension *.sqb*. See SQL Backup File Converter for more information.

If you convert a backup file that was created with multiple threads (THREADCOUNT > 1), an individual .bak file is created for each thread. For example, if backup file *BackUpA.sqb* was created using three threads, the following files are created when you convert the backup:

> *BackUpA_01.bak*
>
> *BackUpA_02.bak*
>
> *BackUpA_03.bak*

You can then restore the MTF (.bak) format backups as a group of backup files.

> ⚠ 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

```
CONVERT { 'p_hysical_backup_device_name' } ] TO { 'physical_file_name' }
[ WITH
  [ [ , ] MAXDATABLOCK = { 65536 | 131072 | ... | 2097152 } ]
  [ [ , ] PASSWORD = { 'password' | 'FILE:file_path'} ]
  [ [ , ] SINGLERESULTSET ]
]
```

Arguments

MAXDATABLOCK

Specifies the maximum size of data blocks to be used when SQL Backup Pro stores backup data. Valid values are integers in multiples of 65536, up to a maximum value of 2097152. For example:

```
MAXDATABLOCK = 655360
```

PASSWORD

Specifies the password to be used with encrypted SQL Backup backup files. The resulting MTF (.bak) format file will not be encrypted.

> ⓘ **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.

SINGLERESULTSET

Specifies that the results returned by the `CONVERT` command should be limited to just one result set. This may be useful if you want to manipulate results using a Transact-SQL script. Such scripts can only manipulate results when a single result set is returned. The `CONVERT` command will return two result sets by default unless you specify the `SINGLERESULTSET` keyword.

Example

This example converts the encrypted backup file *pubs.sqb* to an unencrypted MTF format file:

```
SQLBackupC.exe -I {instance name} -SQL "CONVERT 'C:\Backups\pubs.sqb' TO 'C:\Backups\pubs.bak' WITH PASSWORD =
'MyPassword'"
```

```
EXECUTE master..sqlbackup '-SQL "CONVERT ''C:\Backups\pubs.sqb'' TO ''C:\Backups\pubs.bak'' WITH PASSWORD =
''MyPassword''"'
```