

BitLocker: Use BitLocker Drive Encryption Tools to manage BitLocker

Article • 06/05/2023 •

Applies  [Windows 11](#),  [Windows 10](#),  [Windows Server 2022](#),  [Windows Server 2019](#), 
to: [Windows Server 2016](#)

This article for the IT professional describes how to use tools to manage BitLocker.

BitLocker Drive Encryption Tools include the command-line tools `manage-bde` and `repair-bde` and the BitLocker cmdlets for Windows PowerShell.

Both `manage-bde` and the BitLocker cmdlets can be used to perform any task that can be accomplished through the BitLocker control panel and are appropriate to use for automated deployments and other scripting scenarios.

`Repair-bde` is a special circumstance tool that is provided for disaster recovery scenarios in which a BitLocker protected drive can't be unlocked normally or using the recovery console.

1. [Manage-bde](#)
2. [Repair-bde](#)
3. [BitLocker cmdlets for Windows PowerShell](#)

Manage-bde

`Manage-bde` is a command-line tool that can be used for scripting BitLocker operations. `Manage-bde` offers additional options not displayed in the BitLocker control panel. For a complete list of the `manage-bde.exe` options, see the [Manage-bde](#) command-line reference.

`Manage-bde` includes fewer default settings and requires greater customization for configuring BitLocker. For example, using just the `manage-bde.exe -on` command on a data volume will fully encrypt the volume without any authenticating protectors. A volume encrypted in this manner still requires user interaction to turn on BitLocker protection, even though the command successfully completed because an authentication method needs to be added to the volume for it to be fully protected. The following sections provide examples of common usage scenarios for `manage-bde`.

Using manage-bde with operating system volumes

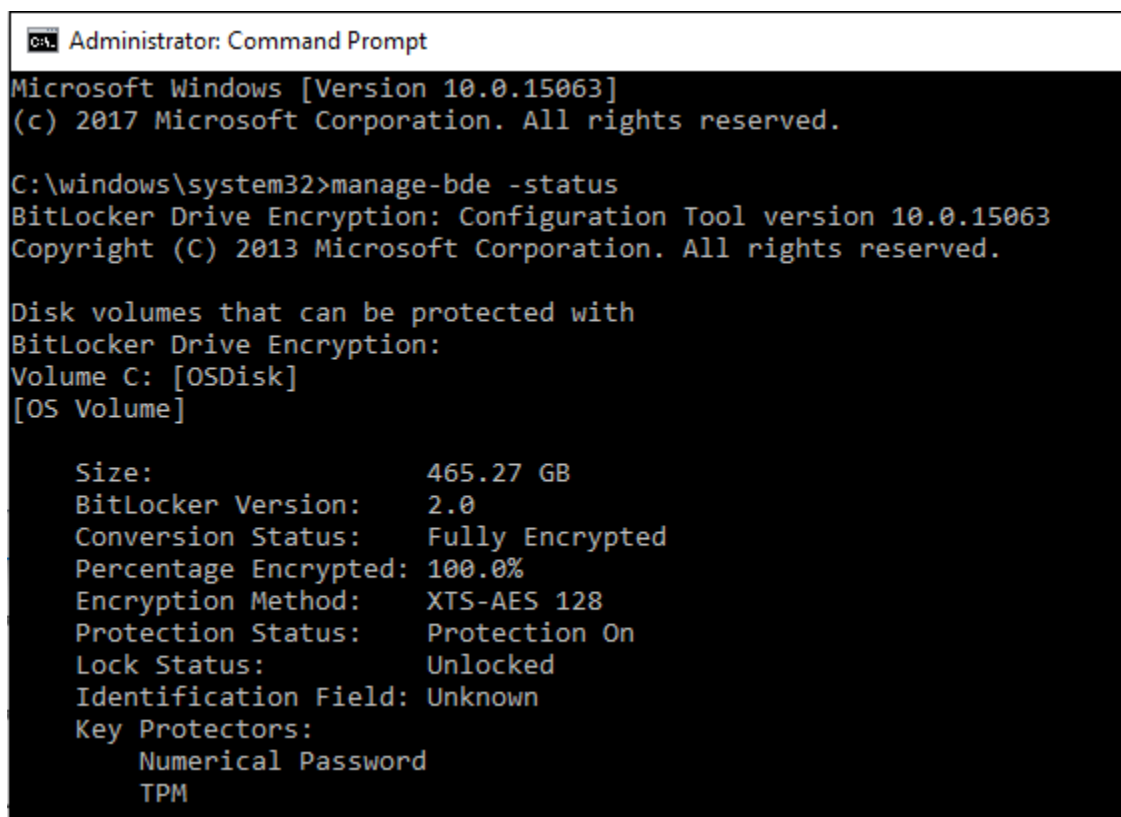
Listed below are examples of basic valid commands for operating system volumes. In general, using only the `manage-bde.exe -on <drive letter>` command will encrypt the operating system volume with a TPM-only protector and no recovery key. However, many environments require more secure protectors such as passwords or PIN and expect information recovery with a recovery key. It's recommended to add at least one primary protector plus a recovery protector to an operating system volume.

A good practice when using `manage-bde.exe` is to determine the volume status on the target system. Use the following command to determine volume status:

Windows Command Prompt

```
manage-bde.exe -status
```

This command returns the volumes on the target, current encryption status, encryption method, and volume type (operating system or data) for each volume:



```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\windows\system32>manage-bde -status
BitLocker Drive Encryption: Configuration Tool version 10.0.15063
Copyright (C) 2013 Microsoft Corporation. All rights reserved.

Disk volumes that can be protected with
BitLocker Drive Encryption:
Volume C: [OSDisk]
[OS Volume]

Size: 465.27 GB
BitLocker Version: 2.0
Conversion Status: Fully Encrypted
Percentage Encrypted: 100.0%
Encryption Method: XTS-AES 128
Protection Status: Protection On
Lock Status: Unlocked
Identification Field: Unknown
Key Protectors:
    Numerical Password
    TPM
```

The following example illustrates enabling BitLocker on a computer without a TPM chip. Before beginning the encryption process, the startup key needed for BitLocker must be

created and saved to a USB drive. When BitLocker is enabled for the operating system volume, BitLocker will need to access the USB flash drive to obtain the encryption key. In this example, the drive letter E represents the USB drive. Once the commands are run, it will prompt to reboot the computer to complete the encryption process.

Windows Command Prompt

```
manage-bde.exe -protectors -add C: -startupkey E:  
manage-bde.exe -on C:
```

Note

After the encryption is completed, the USB startup key must be inserted before the operating system can be started.

An alternative to the startup key protector on non-TPM hardware is to use a password and an **ADaccountorgroup** protector to protect the operating system volume. In this scenario, the protectors are added first. To add the protectors, enter the following command:

Windows Command Prompt

```
manage-bde.exe -protectors -add C: -pw -sid <user or group>
```

The above command will require the password protector to be entered and confirmed before adding them to the volume. With the protectors enabled on the volume, BitLocker can then be turned on.

On computers with a TPM, it's possible to encrypt the operating system volume without defining any protectors using `manage-bde.exe`. To enable BitLocker on a computer with a TPM without defining any protectors, enter the following command:

Windows Command Prompt

```
manage-bde.exe -on C:
```

The above command encrypts the drive using the TPM as the default protector. If verify if a TPM protector is available, the list of protectors available for a volume can be listed

by running the following command:

Windows Command Prompt

```
manage-bde.exe -protectors -get <volume>
```

Using manage-bde with data volumes

Data volumes use the same syntax for encryption as operating system volumes but they don't require protectors for the operation to complete. Encrypting data volumes can be done using the base command:

```
manage-bde.exe -on <drive letter>
```

or additional protectors can be added to the volume first. It's recommended to add at least one primary protector plus a recovery protector to a data volume.

A common protector for a data volume is the password protector. In the example below, a password protector is added to the volume and then BitLocker is turned on.

Windows Command Prompt

```
manage-bde.exe -protectors -add -pw C:  
manage-bde.exe -on C:
```

Repair-bde

Hard disk areas on which BitLocker stores critical information could be damaged, for example, when a hard disk fails or if Windows exits unexpectedly.

The BitLocker Repair Tool (Repair-bde) can be used to access encrypted data on a severely damaged hard disk if the drive was encrypted with BitLocker. Repair-bde can reconstruct critical parts of the drive and salvage recoverable data as long as a valid recovery password or recovery key is used to decrypt the data. If the BitLocker metadata data on the drive has become corrupt, the backup key package in addition to the recovery password or recovery key must be supplied. This key package is backed up in Active Directory Domain Services (AD DS) if the default settings for AD DS backup are used. With this key package and either the recovery password or recovery key, portions

of a corrupted BitLocker-protected drive can be decrypted. Each key package will work only for a drive that has the corresponding drive identifier. The BitLocker Recovery Password Viewer can be used to obtain this key package from AD DS.

Tip

If recovery information is not being backed up to AD DS or if key packages need to be saved in an alternative way, the command:

```
manage-bde.exe -KeyPackage
```

can be used to generate a key package for a volume.

The Repair-bde command-line tool is intended for use when the operating system doesn't start or when the BitLocker Recovery Console can't be started. Use Repair-bde if the following conditions are true:

- The drive has been encrypted using BitLocker Drive Encryption.
- Windows doesn't start, or the BitLocker recovery console can't be started.
- There isn't a backup copy of the data that is contained on the encrypted drive.

Note

Damage to the drive may not be related to BitLocker. Therefore, it is recommended to try other tools to help diagnose and resolve the problem with the drive before using the BitLocker Repair Tool. The Windows Recovery Environment (Windows RE) provides additional options to repair computers.

The following limitations exist for Repair-bde:

- The Repair-bde command-line tool can't repair a drive that failed during the encryption or decryption process.
- The Repair-bde command-line tool assumes that if the drive has any encryption, then the drive has been fully encrypted.

For more information about using repair-bde, see [Repair-bde](#).

BitLocker cmdlets for Windows PowerShell

Windows PowerShell cmdlets provide a new way for administrators to use when working with BitLocker. Using Windows PowerShell's scripting capabilities, administrators can integrate BitLocker options into existing scripts with ease. The list below displays the available BitLocker cmdlets.

Name	Parameters
Add-BitLockerKeyProtector	<ul style="list-style-type: none">• ADAccountOrGroup• ADAccountOrGroupProtector• Confirm• MountPoint• Password• PasswordProtector• Pin• RecoveryKeyPath• RecoveryKeyProtector• RecoveryPassword• RecoveryPasswordProtector• Service• StartupKeyPath• StartupKeyProtector• TpmAndPinAndStartupKeyProtector• TpmAndPinProtector• TpmAndStartupKeyProtector• TpmProtector• WhatIf
Backup-BitLockerKeyProtector	<ul style="list-style-type: none">• Confirm• KeyProtectorId• MountPoint• WhatIf
Disable-BitLocker	<ul style="list-style-type: none">• Confirm• MountPoint• WhatIf
Disable-BitLockerAutoUnlock	<ul style="list-style-type: none">• Confirm• MountPoint• WhatIf
Enable-BitLocker	<ul style="list-style-type: none">• AdAccountOrGroup• AdAccountOrGroupProtector

Name	Parameters
	<ul style="list-style-type: none">• Confirm• EncryptionMethod• HardwareEncryption• Password• PasswordProtector• Pin• RecoveryKeyPath• RecoveryKeyProtector• RecoveryPassword• RecoveryPasswordProtector• Service• SkipHardwareTest• StartupKeyPath• StartupKeyProtector• TpmAndPinAndStartupKeyProtector• TpmAndPinProtector• TpmAndStartupKeyProtector• TpmProtector• UsedSpaceOnly• WhatIf
Enable-BitLockerAutoUnlock	<ul style="list-style-type: none">• Confirm• MountPoint• WhatIf
Get-BitLockerVolume	<ul style="list-style-type: none">• MountPoint
Lock-BitLocker	<ul style="list-style-type: none">• Confirm• ForceDismount• MountPoint• WhatIf
Remove-BitLockerKeyProtector	<ul style="list-style-type: none">• Confirm• KeyProtectorId• MountPoint• WhatIf
Resume-BitLocker	<ul style="list-style-type: none">• Confirm• MountPoint• WhatIf
Suspend-BitLocker	<ul style="list-style-type: none">• Confirm• MountPoint• RebootCount

Name	Parameters
	<ul style="list-style-type: none">• WhatIf
Unlock-BitLocker	<ul style="list-style-type: none">• AdAccountOrGroup• Confirm• MountPoint• Password• RecoveryKeyPath• RecoveryPassword• RecoveryPassword• WhatIf

Similar to `manage-bde`, the Windows PowerShell cmdlets allow configuration beyond the options offered in the control panel. As with `manage-bde`, users need to consider the specific needs of the volume they're encrypting prior to running Windows PowerShell cmdlets.

A good initial step is to determine the current state of the volume(s) on the computer. Determining the current state of the volume(s) can be done using the `Get-BitLockerVolume` cmdlet.

The `Get-BitLockerVolume` cmdlet output gives information on the volume type, protectors, protection status, and other details.

Tip

Occasionally, all protectors may not be shown when using `Get-BitLockerVolume` due to lack of space in the output display. If all of the protectors for a volume are not seen, use the Windows PowerShell pipe command (`|`) to format a full listing of the protectors:

```
Get-BitLockerVolume C: | fl
```

To remove the existing protectors prior to provisioning BitLocker on the volume, use the `Remove-BitLockerKeyProtector` cmdlet. Running this cmdlet requires the GUID associated with the protector to be removed.

A simple script can pipe the values of each `Get-BitLockerVolume` return out to another variable as seen below:

PowerShell

```
$vol = Get-BitLockerVolume  
$keyprotectors = $vol.KeyProtector
```

By using this script, the information in the \$keyprotectors variable can be displayed to determine the GUID for each protector.

By using this information, the key protector for a specific volume can be removed using the command:

PowerShell

```
Remove-BitLockerKeyProtector <volume>: -KeyProtectorID "{GUID}"
```

ⓘ Note

The BitLocker cmdlet requires the key protector GUID enclosed in quotation marks to execute. Ensure the entire GUID, with braces, is included in the command.

Using the BitLocker Windows PowerShell cmdlets with operating system volumes

Using the BitLocker Windows PowerShell cmdlets is similar to working with the manage-bde tool for encrypting operating system volumes. Windows PowerShell offers users flexibility. For example, users can add the desired protector as part command for encrypting the volume. Below are examples of common user scenarios and steps to accomplish them in BitLocker Windows PowerShell.

The following example shows how to enable BitLocker on an operating system drive using only the TPM protector:

PowerShell

```
Enable-BitLocker C:
```

In the example below, adds one additional protector, the StartupKey protector and chooses to skip the BitLocker hardware test. In this example, encryption starts

immediately without the need for a reboot.

PowerShell

```
Enable-BitLocker C: -StartupKeyProtector -StartupKeyPath <path>  
-SkipHardwareTest
```

Using the BitLocker Windows PowerShell cmdlets with data volumes

Data volume encryption using Windows PowerShell is the same as for operating system volumes. Add the desired protectors prior to encrypting the volume. The following example adds a password protector to the E: volume using the variable \$pw as the password. The \$pw variable is held as a SecureString value to store the user-defined password.

PowerShell

```
$pw = Read-Host -AsSecureString  
<user inputs password>  
Enable-BitLockerKeyProtector E: -PasswordProtector -Password $pw
```

Using an AD Account or Group protector in Windows PowerShell

The **ADAccountOrGroup** protector, introduced in Windows 8 and Windows Server 2012, is an Active Directory SID-based protector. This protector can be added to both operating system and data volumes, although it doesn't unlock operating system volumes in the pre-boot environment. The protector requires the SID for the domain account or group to link with the protector. BitLocker can protect a cluster-aware disk by adding a SID-based protector for the Cluster Name Object (CNO) that lets the disk properly fail over to and become unlocked by any member computer of the cluster.

Warning

The **ADAccountOrGroup** protector requires the use of an additional protector for use (such as TPM, PIN, or recovery key) when used on operating system volumes

To add an **ADAccountOrGroup** protector to a volume, use either the actual domain SID or the group name preceded by the domain and a backslash. In the example below, the `CONTOSO\Administrator` account is added as a protector to the data volume G.

PowerShell

```
Enable-BitLocker G: -AdAccountOrGroupProtector -AdAccountOrGroup  
CONTOSO\Administrator
```

For users who wish to use the SID for the account or group, the first step is to determine the SID associated with the account. To get the specific SID for a user account in Windows PowerShell, use the following command:

Note

Use of this command requires the RSAT-AD-PowerShell feature.

PowerShell

```
get-aduser -filter {samaccountname -eq "administrator"}
```

Tip

In addition to the PowerShell command above, information about the locally logged on user and group membership can be found using: `WHOAMI /ALL`. This doesn't require the use of additional features.

The following example adds an **ADAccountOrGroup** protector to the previously encrypted operating system volume using the SID of the account:

PowerShell

```
Add-BitLockerKeyProtector C: -ADAccountOrGroupProtector  
-ADAccountOrGroup S-1-5-21-3651336348-8937238915-291003330-500
```

Note

Active Directory-based protectors are normally used to unlock Failover Cluster enabled volumes.

Related articles

- [BitLocker overview](#)
- [BitLocker frequently asked questions \(FAQ\)](#)
- [Prepare your organization for BitLocker: Planning and policies](#)
- [BitLocker: How to enable Network Unlock](#)
- [BitLocker: How to deploy on Windows Server 2012](#)