BitLocker Drive Encryption: EnableBitLocker.vbs Overview and Parameter Reference

Microsoft Corporation

First Published: September 2009

Last Updated: December 2010

Abstract

This document includes an overview of the EnableBitLOcker.vbs sample WMI script and details the different functions within the script. This document also includes a parameter reference which you can use as a reference when testing the script at a command prompt.

The document was last updated to fix an error in the parameter reference that misidentified the parameter used to enable BitLocker using the **TPM + PIN** authentication method.

This is a preliminary document and may be changed substantially prior to final commercial release of the software described herein.

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in examples herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

Contents

EnableBitLocker.vbs Overview and Parameter Reference 4
EnableBitLocker.vbs overview 4
Detailed script information 5
Section: General 1 5
Section: General 2 5
Function 1: Connect to the TPM WMI provider 7
Function 2: Connect to the BitLocker WMI provider 7
Function 3: Get BitLocker status data and enable encryption 7
Function 4: Get the TPM status 8
Function 5: Enable and activate the TPM 8
Function 6: Create a recovery key 8
Function 7: Change TPM owner information 8
Function 8: Generate a random string for the TPM password 9
Function 9: Take ownership of the TPM 9
Function 10: Enable BitLocker 9
Function 11: Find removable drive 10
Functions 12 and 13: Interrogate BitLocker Group Policy and verify backup 10
Function 14: Create SMS status MIF 11
Function 15: Check for logged-on user 11
Function 16: Create recovery password 11
Function 17: Request PIN from user 11
Function 18: Determine conversion status 11
Function 19: Check protection errors 12
Function 20: Help 12
EnableBitLocker.vbs parameter reference 12
Required commands 12
/on 12
/l 13
Optional commands 14
/em 14
/rk 15
/ro 15

EnableBitLocker.vbs Overview and Parameter Reference

You can use the EnableBitLocker.vbs sample script as an example of how to automate the deployment and configuration of BitLocker Drive Encryption. The script is fully functional, but you may need to customize certain aspects of it to meet your organization’s needs.

DISCLAIMER

The sample scripts are not supported under any Microsoft standard support program or service. The sample scripts are provided AS IS without warranty of any kind. Microsoft further disclaims all implied warranties including, without limitation, any implied warranties of merchantability or of fitness for a particular purpose. The entire risk arising out of the use or performance of the sample scripts and documentation remains with you. In no event shall Microsoft, its authors, or anyone else involved in the creation, production, or delivery of the scripts be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use of or inability to use the sample scripts or documentation, even if Microsoft has been advised of the possibility of such damages.

- EnableBitLocker.vbs overview
- EnableBitLocker.vbs parameter reference

EnableBitLocker.vbs overview

This sample script is designed to be used for all BitLocker configuration scenarios. It can be run multiple times on a computer. The script automates the following BitLocker configuration settings.

- Enable and activate the TPM
- Take ownership of the TPM and generate random owner password
- Enable BitLocker protection using
  - TPM only
  - TPM and PIN
  - TPM and Startup Key
  - USB only
- Create additional recovery key
- Create recovery password
- Specify encryption method
- Reset TPM owner information

How you choose to implement the script depends on how you deploy desktops in your environment. This script is designed to work with the following deployment tools.

- Group Policy startup and logon scripts

- Windows Vista and Windows 7 setup first logon commands
- SMS 2003 software distribution
- Business Desktop Deployment 2007 automation
- Other software distribution tools

Important

You can use a WMI script to create System Management Server (SMS) status Management Information Format (MIF) files. To create status MIFs on the SMS client during software distribution, two DLLs must be registered on the client itself. For more information and detailed instructions on how to create status MIF files in SMS, see How to Create Status MIF Files in SMS (http://go.microsoft.com/fwlink/?LinkId=93539).

Detailed script information

This document describes a sample script named EnableBitLocker.vbs. The following sections describe how the parts of the script work and the function of each. Use this information to customize the script for your specific deployment needs.

Section: General 1

This section describes the part of the script that prepares the computer for BitLocker.

- Create the file system and shell objects used throughout the script.
- Create the log file for script processing.
- Evaluate the command-line arguments supplied by the user.

This section is most important for evaluating the command-line argument supplied to the script.

Section: General 2

This is the main processing section of the script. Initially functions are called to make the connection to the TPM, get the status of the TPM, and make the connection to the BitLocker provider. Then several additional functions use this data to enable BitLocker. Table 1 shows the TPM and BitLocker functions referenced in the General 2 section of the script.

Table 1 General 2 functions in EnableBitLocker.vbs

Function	Description
ConnectTPMProv()	Connects to the TPM WMI provider.
GetTPMStatus()	Gets the current status of the TPM.
ConnectBDEProv()	Connects to the volume encryption WMI provider.

Function	**Description**
GetBDEStatus()	Gets the BitLocker status data and enables encryption.
EnableActivateTPM()	Enables and activates the TPM.
CreateRK(sProtID)	Creates a recovery key, if the parameter is specified when the script is run.
ChangeOwnerAuth(strOldOwnerPassword,strOwnerPassword)	Changes the ownership of a previously-owned TPM, if the parameter is specified when the script is run.
DenTPMPassword	Generates a random string for the TPM owner password.
OwnTPM	Takes ownership of the TPM.
EnableBitlocker(objEnVol)	Enables BitLocker Drive Encryption.
FindRemovableDrive()	Uses WMI to determine whether there is a USB flash drive available to the system.
EvalGPO and getBDEEvents()	Determines if BitLocker Group Policy is applied to the local computer.
CreateStatusMIF(strStatusData)	Applies when the /SMS parameter is used on the command line. Using this option allows you to add information to the SMS status messaging during an SMS software distribution.
CheckUser	Queries WMI for current logged-on user information.
CreateRP(objEnVol)	Creates a recovery password for BitLocker using the BitLocker WMI provider method ProtectKeyWithNumericalPassword.
GetPIN	Prompts the user of the computer to enter a PIN for BitLocker. This

Function	Description
	function is called only when /on:tp is used.
GetConversionStatus(strCS)	Checks whether BitLocker is already enabled.
CheckError(intRC)	Checks for common errors that may occur when ProtectWith functions are called in Function 3.
ShowHelp	Displays Help if incorrect command line options are used.

Function 1: Connect to the TPM WMI provider

This function, ConnectTPMProv(), connects to the TPM Provider class and tries to retrieve the TPM instance. If an instance of the TPM provider class is not present and the protect option is set for USB, then the script will continue. If no instance of the TPM provider class is present and the USB option is not selected, then the script will exit with a failure code.

Function 2: Connect to the BitLocker WMI provider

This function, ConnectBDEProv(), connects to the BitLocker (MicrosoftVolumeEncryption) Provider class. If the connection fails, then the script exits with a failure code.

Function 3: Get BitLocker status data and enable encryption

This function, GetBDEStatus(), determines BitLocker status, sets the requested Key Protectors, and then initiates encryption. To accomplish these tasks, the script determines the count of volumes that can be encrypted and then determines the correct volume to encrypt by identifying the WMI operating system volume. After it identifies the correct volume, the script retrieves the current BitLocker Protection status. If BitLocker is already enabled, the script will exit with a failure code. If BitLocker is not enabled, the script will determine the current conversion state. It is possible to have the following conversion states:

- The volume has a status of fully decrypted.
- The volume has a status of fully encrypted but a clear key is present.
- The volume has a status of encryption in progress.
- The volume has a status of decryption in progress.
- The volume has a status of encryption paused.
- The volume has a status of decryption paused.

If a conversion state of “fully decrypted” is detected, the requested Key Protectors are enabled and encryption begins. If the conversion state is “fully encrypted but a clear key is present” then the clear key is removed. If any other conversion state is detected, the system is classified as being in a transient state and the script will exit with a failure code. Table 2 shows the sub-functions that GetBDEStatus() calls.

Table 2   GetBDEStatus() sub-functions

Function	Description
CreateStatusMIF	Creates an SMS status MIF if the /SMS command was used.
CreateRP	Creates a protector key with a numeric password.
EnableBitLocker	Initiates BitLocker drive encryption.
FindRemovableDrive()	Finds removable drive(s).

Function 4: Get the TPM status

This function, GetTPMStatus(), determines the status of the TPM. The possible TPM states are:

- Enabled
- Activated
- Owned

Function 5: Enable and activate the TPM

This function, EnableActivateTPM, sets the TPM to enable and activate. Once the TPM is set, the script determines if a reboot or shutdown is required to complete the action. The proper action is then executed.

Note
Physical presence is normally required after the reboot / shutdown to complete the enable and activate action.

Function 6: Create a recovery key

If specified by a command-line parameter, this function, CreateRK(sProtID), creates a recovery key and saves it to a USB flash drive. The function calls the FindRemovableDrive function to determine the USB drive letter. After the drive letter is established, the recovery key is created and saved to the USB drive.

Function 7: Change TPM owner information

If specified by a command-line parameter, this function, ChangeOwnerAuth(strOldOwnerPassword,strOwnerPassword), changes the ownership of a previously-owned TPM. This function requires that the old TPM owner password be entered in the command line. The TPM owner password is changed to the 20-byte owner authorization that the ChangeOwnerAuth method requires. The DenTPMPassword function generates the new TPM owner password and converts it to the 20-byte owner authorization format. After the old and new passphrases have been converted to the new format, the ChangeOwnerAuth method is used to change TPM ownership.

Function 8: Generate a random string for the TPM password

This function, DenTPMPassword, generates a string for use as the TPM password. The password will be 7 to 14 characters long and will contain both letters and numbers. The random password generated in this function is passed to the OwnTPM function.

> **Note**
>
> This function writes the owner password into the script log. This is done only as an example. Remove this line in the script before using in production. Use Group Policy to force backup of the TPM owner password in Active Directory Domain Services.

Function 9: Take ownership of the TPM

This function, OwnTPM, takes ownership of the TPM and verifies that there is an endorsement key on the TPM. If there is no endorsement key, one will be created before the take ownership process is started. If an endorsement key is present, this step is skipped. This function uses output from the random password generated in DenTPMPassword function and hashes the passphrase to the 20-byte owner authorization used by the TakeOwnership method. The 20-byte owner authorization is then used to take ownership of the TPM.

Function 10: Enable BitLocker

This function, EnableBitlocker(objEnVol), enables BitLocker Drive Encryption. If all preconditions are satisfied, the script initiates drive encryption using the selected encryption key size. Select one of the following key sizes:

- Unspecified

  Will use the settings in Group Policy, otherwise will use the default.
- AES 128 with Diffuser (default setting)
- AES 256 with Diffuser (as stated)
- AES 128 (as stated)
- AES 256 (as stated)

This function also checks for the existence of the Group Policy registry value that is used to set a mandatory encryption method. If this value exists, it will be used instead of any /em arguments that are specified on the command line. The registry value can be found in the following location:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\FVE
```

Function 11: Find removable drive

This function, FindRemovableDrive(), uses WMI to determine whether a USB flash drive is available to the system. Several other functions in the script also use this function if specific command-line options are used. Table 3 shows these command-line options.

Table 3   FindRemovableDrive() usage with the command-line

Parameters	Description
/on:tsk /promptuser	Protect with TPM and startup key.
/rk /promptuser	Create a recovery key.

If you specify either of the options in Table 3, a USB flash drive must be available or the function will prompt the user to insert one. The function checks whether a USB is available or if more than one USB is available. If more than one USB device is available, the function stops the script with a failure exit code and logs the reason for failure. If a user does not enter a valid USB within three attempts, the script will quit with a failure exit code, in order to prohibit an infinite loop in this process.

Functions 12 and 13: Interrogate BitLocker Group Policy and verify backup

These functions, EvalGPO and getBDEEvents(), determine whether BitLocker Group Policy is applied to the local computer. The WMI StdRegProv queries the registry. After the connection is made to the registry, (`HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft`) is checked for the existence of the FVE key. If this key exists, then the keys beneath the FVE key are enumerated along with the values associated with them. These values are then stored to variables for use later in the script. Table 4 shows the configuration of BitLocker Group Policy settings associated to the registry keys.

Table 4   Group Policy Settings

Configuration	Registry values

Client Group Policy configured to require AD DS escrow of recovery password.	ActiveDirectoryBackup = 1
	RequireActiveDirectoryBackup = 1

Client Group Policy configured to require AD DS escrow of recovery password and key packages.	ActiveDirectoryBackup = 1
	RequireActiveDirectoryBackup = 1

Client Group Policy is configured to require AD DS escrow of recovery password but is not mandatory.	ActiveDirectoryBackup = 1
	RequireActiveDirectoryBackup = 0

Client Group Policy is configured to require AD DS escrow of recovery passwords and key packages but is not mandatory.	ActiveDirectoryBackup = 1
	RequireActiveDirectoryBackup = 0

Function 14: Create SMS status MIF

This function, CreateStatusMIF(strStatusData), applies when the /SMS option is used on the command line. Using this option allows you to add information to the SMS status messaging during an SMS software distribution. This requires additional DLLs to be registered on the SMS Advanced Client computer. For more information on the additional DLL requirements, see How to Create Status MIF Files in SMS (http://go.microsoft.com/fwlink/?LinkId=93539).

Function 15: Check for logged-on user

This function, CheckUser, queries WMI for current logged-on user information. This information is retrieved from the Win32_ComputerSystem class. If a user is logged in, the strCurrentUser variable is set to 1 and is used later in the script when creating PINs, startup keys, and recovery keys.

Function 16: Create recovery password

This function, CreateRP(objEnVol), creates a recovery password for BitLocker using the BitLocker WMI provider method ProtectKeyWithNumericalPassword. This function is always run regardless of which /ON option is used to enable BitLocker. If BitLocker is configured to back up recovery information to Active Directory Domain Services (AD DS), backup will occur when this method is used. If AD DS backup is required and this function fails because it cannot contact AD DS, then all of the protectors that were previously created will be deleted, and the script will exit with a failure code. If AD DS backup is not required, the script will query the event log for event IDs 513 and 514. Event ID 514 means a successful backup of the computer recovery information to AD DS. Event ID 513 means a failure to back up computer recovery information to Active Directory. If either of these events is found, it is sent to the script log.

If the script does not find the FVE policy key on the system, no Group Policy is configured and the event log will not be scanned after the recovery password is created.

Function 17: Request PIN from user

This function, GetPIN, prompts the user of the computer to enter a PIN for BitLocker. This function is called only when /on:tp is used. After the user enters a PIN, it is checked to make sure it meets the necessary requirements. The PIN must be between 4 and 20 numeric digits long. If the PIN does not meet this requirement, the user is prompted with a dialog telling him or her what was incorrect with the PIN and is given another chance to enter a valid PIN. If the user fails three times to enter a valid PIN, the script will log this as a critical failure and exit with a failure.

Function 18: Determine conversion status

This function, GetConversionStatus(strCS), is used during the GetBDEStatus function if BitLocker is already enabled. The script checks whether BitLocker is already enabled. If BitLocker is already enabled, the function discovers the encryption status. If the operating volume is fully encrypted but BitLocker is disabled (a clear key is present on the computer), all key protectors are enabled, causing BitLocker to protect the computer. If the configuration is in any other state, the data is logged and the script exits.

Function 19: Check protection errors

This function, CheckError(intRC), checks for common errors that may occur when ProtectWith functions are called in Function 3. The first error that is checked for is 80310030, which means that the computer contains a bootable CD, DVD, or USB device. This error is critical and causes the script to stop with a failure exit code.

Function 20: Help

This function, ShowHelp, displays Help if incorrect command line options are used.

EnableBitLocker.vbs parameter reference

The following table shows the formatting legend for the command syntax.

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (…)	Parameter that can be repeated several times in a command line
Between brackets([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	A set of choices from which the user must choose only one

Required commands

You must specify both the /on and /l commands when using the EnableBitLocker.vbs script:

EnableBitLocker.vbs /on:{**tpm** | **tp** | **tsk** | **usb**} [**/promptuser**] **/l:***Location*

/on

Syntax	**EnableBitLocker.vbs /on:**{**tpm** | **tp** | **tsk** | **usb**} [**/promptuser**] **/l:***Location*

Parameters	• **tpm** Enables Bitlocker using the TPM only authentication method. • **tp** Enables BitLocker using the TPM + PIN authentication method. You must include the **/promptuser** parameter with **tp**. • **tsk** Enables BitLocker using the TPM + startup key authentication method. You must include the **/promptuser** parameter with **tsk**. • **usb** Enables BitLocker using the startup key (stored on a USB flash drive) only authentication method. You must include the **/promptuser** parameter with **usb**. • **/promptuser** Prompts the user either to enter a PIN number or to provide a USB flash drive, if no USB flash drive is attached to the computer. This parameter must be used with the **tp**, **tsk**, and **usb** parameters.
Remarks	Required. Specifies what authentication method to use for BitLocker. You must select one of the authentication method parameters only: **tpm**, **tp**, **tsk**, or **usb**.
Example command	• To enable BitLocker with the TPM only authentication mode: **EnableBitLocker.vbs /on:tpm /l:**C:\BitLocker.log • To enable BitLocker with the startup key only authentication mode: **EnableBitLocker.vbs /on:usb /promptuser /l:**C:\BitLocker.log

/l

Syntax	**EnableBitLocker.vbs /on:**{**tpm | tp | tsk | usb**} [**/promptuser**] **/l:***Location*
Parameters	*Location* Specifies the location and name for the BitLocker log file.
Remarks	Required. Specifies the location and name for the BitLocker log file.
Example command	To enable BitLocker with the startup key only authentication mode and specify that the log file should be created at C:\BitLocker.log: **EnableBitLocker.vbs /on:usb /promptuser /l:**C:\BitLocker.log

Optional commands

In addition to the required commands above, you can specify the additional optional commands:

EnableBitLocker.vbs /on:{**tpm | tp | tsk | usb**} [**/promptuser**] **/l:***Location* [**/em:**{**128d | 256d \ 128 | 256**}] [**/rk:***PathtoExternalKeyDirectory*] [**/ro:***ExistingOwnerPassword*]

/em

Syntax	**EnableBitLocker.vbs /on:**{**tpm | tp | tsk | usb**} [**/promptuser**] **/l:***Location* [**/em:**{**128d | 256d \ 128 | 256**}]
Parameters	• **128d** Specifies that 28-bit AES encryption should be used with the diffuser. • **256d** Specifies that 256-bit AES encryption should be used with the diffuser. • **128** Specifies that 128-bit AES encryption should be used. • **256** Specifies that 256-bit AES encryption should be used.
Remarks	Specifies the encryption method to use to encrypt the hard disk. If you do not use this

	command, then the script will use 128-bit with the diffuser by default.
Example command	To use 256-bit encryption with the diffuser: **EnableBitLocker.vbs /on:tpm /l:**C:\BitLocker.log **/em:**256d

/rk

Syntax	**EnableBitLocker.vbs /on:**{**tpm | tp | tsk | usb**} [**/promptuser**] **/l:***Location* [**/rk:***PathtoExternalKeyDirectory*]
Parameters	*PathtoExternalKeyDirectory* Specifies the location of the external USB flash drive.
Remarks	Creates a recovery key on an available USB flash drive.
Example command	To create a recovery key on a USB flash drive: **EnableBitLocker.vbs /on:tpm /l:**C:\BitLocker.log /**rk:**F:\RK\

/ro

Syntax	**EnableBitLocker.vbs /on:**{**tpm | tp | tsk | usb**} [**/promptuser**] **/l:***Location* [**/ro:***ExistingOwnerPassword*]
Parameters	*ExistingOwnerPassword* Specifies the existing TPM owner password.
Remarks	Changes the existing TPM owner password to a randomly generated password. You must enter the existing TPM owner password in order to change it.
Example command	To change the existing TPM owner password to a randomly generated password: **EnableBitLocker.vbs /on:tpm /l:**C:\BitLocker.log **/ro:**"28998237487"