



IRONKEY™ WORKSPACE

Scripting Guide

Using scripts to provision and deploy Windows To Go drives

Copyright © 2015 Imation Corp. All rights reserved.

Imation and Imation logo, IronKey and IronKey logo, and “PC on a Stick” are trademarks of Imation Corp. All other trademarks are the property of their respective owners.

Imation Enterprises Corp.

1 Imation Way

Oakdale, MN 55128-3414 USA

www.imation.com

10/15 IK-PROV-ADM02-1.1

NOTE: Imation is not liable for technical or editorial errors and/or omissions contained herein; nor for incidental or consequential damages resulting from the furnishing or use of this material. The information provided herein is subject to change without notice.

The information contained in this document represents the current view of Imation on the issue discussed as of the date of publication. Imation cannot guarantee the accuracy of any information presented after the date of publication. This document is for information purposes only. Imation makes no warranties, expressed or implied, in this document.



CONTENTS

Introduction	4
Who should read this guide?	4
About Windows To Go	4
About IronKeyUtil	5
Supported devices	5
Requirements for provisioning devices	5
Imation Documentation	7
Getting started	8
Setup requirements	8
General steps to include in provisioning scripts	8
IronKeyUtil commands	11
IronKeyUtil command line Help (?)	11
Documentation conventions	13
init command	13
check_os command	16
apply_mode command	18
Logging and error codes	20
Log file location and content	20
Troubleshooting error messages	21
Provisioning devices using PowerShell scripts	23
Requirements	23
What's in the ZIP file?	24
Sample PowerShell script	25
Before you run the script	26
Running the PowerShell script	28
Provisioning devices using batch scripts	31
Requirements	31
What's in the ZIP file?	32
Sample batch provisioning scripts	33
Before you run the script	38
Running the sample batch script	39
Provisioning devices using SCCM	42
Troubleshooting	43

INTRODUCTION

The IronKey Workspace Scripting Guide provides an overview of how to provision IronKey Workspace devices using scripts. With the IronKey Workspace Command Line Utility (IronKeyUtil), you can now provision hardware-encrypted W500/W700 devices in a scripted work flow. Included in the release package are sample PowerShell and batch provisioning scripts to get you started. The scripts are meant to be modified to meet your provisioning requirements and environment. The scripts will also provision W300 and W500/W700 drives. For W500/W700 devices, the sample scripts incorporate important IronKeyUtil commands to initialize and prepare these devices for deployment.

Using scripts to provision Windows To Go drives reduces the cycle time when deploying large volumes of devices. Script files are portable so you can run the same provisioning script on different computers in parallel. The sample scripts can also be deployed through SCCM to create Windows To Go drives.

This chapter includes information about the following topics:

- Who should read this guide?
- About Windows To Go
- About IronKeyUtil
- Supported devices
- Requirements for provisioning devices
- Imation Documentation

WHO SHOULD READ THIS GUIDE?

This guide is written for anyone who creates and deploys Windows laptops and desktops and is tasked with provisioning IronKey Workspace devices. This guide is intended for users who are familiar with scripting and command line tools and interfaces. It describes how to use the IronKey Workspace Command Line Utility to initialize and prepare W500/W700 drives for provisioning and deployment to users. It also incorporates how to use the tool in sample provisioning scripts. The sample scripts can be used with minimal modifications to perform basic provisioning. More advanced deployments may require further customization to meet the unique needs of your provisioning and deployment environment.

For a general overview of Windows To Go and the process involved when planning, deploying and managing IronKey Workspace devices, see the *IronKey Workspace IT Administrators Handbook*, which can be found on the *IronKey Support site*.

ABOUT WINDOWS TO GO

An enterprise feature of Windows 8.1, Windows To Go is a fully manageable Windows 8.1 operating system that you can boot and run from an IronKey Workspace USB device on host computers that meet Windows 7 (or higher) certification requirements and qualified Mac computers.

ABOUT IRONKEYUTIL

The IronKey Workspace Command Line Utility (IronKeyUtil) is an application that is required when provisioning W500/W700 devices using scripts. These devices use hardware encryption to protect the operating system partition. They must be initialized using IronKeyUtil to unlock the secure OS partition. Once unlocked, they can be provisioned with a Windows To Go image. IronKeyUtil (and supporting files) are included in the script package.

This application is not required to provision W300 devices. Once you specify the location of your WIM image, these devices can be provisioned out-of-the-box using the sample scripts included in the package.

SUPPORTED DEVICES

IronKeyUtil version 1.5 provides software and sample scripts that support provisioning IronKey Workspace W300, W500, and W700 devices. Hardware encrypted devices (W500/W700) must be at version 4.3. Devices that are running an earlier version must be upgraded in order to use the IronKey Workspace Command Line Utility. See “IronKeyUtil **script** package (ironkeyutil1.5.zip)” on page 6.

Updates for IronKey Workspace devices are available on the *IronKey Support site*.

Figure 1-1: IronKey Workspace devices



IronKey Workspace devices are trusted, secure USB flash drives. As Microsoft-certified Windows To Go devices, they allow users to change virtually any computer into their own secure personal workspace, capable of using all hardware resources on the host computer, without ever accessing the host computer hard drive; the Windows To Go workspace is isolated from any malware on the host operating system. Administrators can control the corporate IT Windows image that installs on the device to include company applications, security controls, VPN access, and more.

IronKey Workspace W500 and W700 devices use hardware encryption to secure the operating system partition; the W700 model meets the highest security requirements as it is certified to FIPS-140-02 Level 3. These devices can be managed using an IronKey Enterprise Management System (EMS) if licensed, either IronKey Enterprise Service or IronKey Enterprise Server. W300 devices do not support Enterprise management at this time.

Devices that are *New* from manufacturing are ready to provision. W500/W700 devices that are currently in the field must be *Recommissioned*, if managed and activated by EMS, or *Reset* if unmanaged, before they can be re-provisioned for another user. W300 devices can be re-provisioned at any time in any state.

REQUIREMENTS FOR PROVISIONING DEVICES

Hardware

Provisioning computer—A laptop or desktop running Microsoft® Windows® 8.1 Enterprise

IronKeyUtil and the accompanying sample scripts also support using Microsoft Windows 10 Enterprise Technical Preview edition.

Windows 8.1 Enterprise requirements (from <https://technet.microsoft.com/en-us/windows/dn140267.aspx>):

- **Processor**—1 gigahertz (GHz) or faster
- **RAM**—1 gigabyte (GB) (32-bit) or 2 GB (64-bit)
- **Hard disk space**—16 GB (32-bit) or 20 GB (64-bit)
- **Graphics card**—Microsoft DirectX 9 graphics device with WDDM driver
- For 64-bit installations of Windows 8.1 Enterprise, your CPU must also support CompareExchange16b, PrefetchW and LAHF/SAHF
- **USB 3.0 Ports**—Minimum 2 ports (USB 2.0 will work but may be slower)
- **Windows 10 requirement**—Screen resolution: 1024 x 768

Note: The sample scripts require Windows 8.1 Enterprise. Early releases and the Technical Preview version of Windows 10 have been tested with IronKeyUtil and our sample scripts. The sample scripts will require minor modification to remove the requirement that the requires Windows 8.1 Enterprise. See support.ironkey.com for additional information or contact technical support with specific questions.

USB hubs with power adapter

It is recommended that you use USB 3.0 powered hubs. USB 2.0 hubs should work but will be slower. The following USB 3.0 powered hubs are supported with IronKey Workspace devices. This is not an exclusive list and other hubs may be used.

Recommended

- Dyconn USB 3.0 7- port hub, model number HUB7B, or
- Plugable USB 3.0 7-port hub, model number USB3-HUB7-81X

Alternate

- Anker USB 3.0 7-port hub, model number H7928-U3 (with 7 ports of A type output), or
- UtechSmart USB 3.0 7-port hub, model number US-USB3-HUB7

Software

The following requirements are necessary to provision IronKey Workspace devices using scripts.

Windows To Go WIM image

You will need a Microsoft Windows To Go image file (WIM). If OS of the provisioning computer must match the OS of the WIM being deployed. For example, if you are deploying a Windows 8 Enterprise WIM, the host OS must also use Windows 8.1 Enterprise. The WIM should include software and drivers that are required by your end users, including Boot Camp packages if devices will be used on qualified Mac host computers.

IronKeyUtil script package (ironkeyutil1.5.zip)

The IronKeyUtil 1.5 package includes IronKeyUtil and IronKey Control Panel applications and supporting files as well as sample scripts to provision IronKey Workspace devices

- **IronKey Workspace Command Line Utility**—IronKeyUtil is required to initialize and unlock hardware encrypted W500 and W700 devices. Devices must be at version 4.3. For details about specific IronKeyUtil commands, see “IronKeyUtil commands” on page 11.
- **IronKey Control Panel application**— IronKey Control Panel is required in Windows To Go to allow users to manage their device passwords and view device information. Managed devices use the Control Panel to receive commands and updates from the IronKey Enterprise Management System

(EMS), for example, to receive device policy updates or to download and install device upgrade packages. The application must be installed to the OS partition of W500/W700 drives during device provisioning. You can also install this application to the WIM image that will be deployed on devices.

The IronKey Control Panel is available in the *ironkeyutil1.5.zip* package as part of the sample script ZIP files. The sample scripts include command examples that will load IronKey Control Panel on the device at run-time. Do not install the Control Panel to IronKey Workspace W300 devices.

- **Documentation**—This folder includes the End User License Agreement and the Admin Guide (PDF) you are currently reading.
- **Sample scripts**—This folder includes two script packages for provisioning IronKey Workspace devices: Windows Batch scripts and PowerShell scripts. Each package contains scripts for provisioning devices. See “Provisioning devices using batch scripts” on page 31 and “Provisioning devices using PowerShell scripts” on page 23.

IMATION DOCUMENTATION

In addition to this guide, the following documents are available from the support Web site at <http://support.ironkey.com>.

IronKey Workspace

- *IronKey Workspace IT Administrator Handbook*
- *IronKey Workspace W500 User Guide*
- *IronKey Workspace W700 User Guide*

IronKey Enterprise Management System

If your company is using IronKey Enterprise Server or IronKey Enterprise Management Service to manage your deployment of W500 or W700 devices, the following guides provide information on how to setup and use the Server or Service.

IronKey Enterprise Server

- *IronKey Enterprise Server Quick Start Guide*
- *IronKey Enterprise Server Setup Guide*
- *IronKey Enterprise Server Admin Guide*

IronKey Enterprise Management Service

- *IronKey Enterprise Management Service Admin Guide*

GETTING STARTED

Everything you need to create and use scripts to provision IronKey Workspace devices is included in the *ironkeyutil1.5.zip* file. The zipped package contains two other zipped files, the sample batch and sample PowerShell scripts with the supporting files required to run these scripts.

The following lists the contents of the *ironkeyutil1.5.zip* file, including sub-folders and files.

- **ironkeyutil.exe**—This is the executable file that runs IronKey Workspace Command Line Utility (IronKeyUtil). IronKeyUtil is required to initialize and unlock hardware encrypted W500 and W700 devices. For details about specific IronKeyUtil commands, see “IronKeyUtil commands” on page 11.
- **DLL files**—These files are dependencies that are required to run *ironkeyutil.exe* (globalplatform.dll, libeay32.dll, msvcrt100.dll, msxcr100.dll, zlib1.dll).
- **documentation folder**—This includes the End User License agreement and a copy of this Admin Guide (*IronKeyUtil_EULA.html* and *IronKeyUtil_AdminGuide.pdf*).
- **sample_scripts folder**—Contains sample provisioning scripts (*IronKey_batch_sample_scripts.zip* and *IronKey_PowerShell_sample_scripts.zip*). These scripts are meant to be used as a starting point from which you can build your own scripts so that your provisioned devices meet your corporate requirements. Each ZIP file includes a copy of ironkeyutil.exe as well as documentation and the required IronKey Control Panel setup files.

SETUP REQUIREMENTS

- Extract the *ironkeyutil1.5.zip* file to a location that will be accessible from the provisioning computer. You should also extract the sample script files that you plan to use.
- If using USB powered hubs, insert the hubs into USB 3.0 ports (recommended) and plug in the power adapter to a power source. Hubs are not required, you can provision devices using the USB ports (3.0 recommended) on the host system.

GENERAL STEPS TO INCLUDE IN PROVISIONING SCRIPTS

Once you complete the setup tasks to extract the zip file, you are ready to begin the provisioning process. The first step is to create a new script or modify the sample script. If you are provisioning W500/W700 devices, your script must include code to perform the following steps required for these devices:

- **Initialize devices**—This step must be completed before you apply the WIM image to the device. It is required to unlock the encrypted partition on the device in preparation for installing the WIM. Use the IronKeyUtil application, *init command*.
- **Install IronKey Control Panel application to the device**—Once you’ve unlocked the device, you must install IronKey Control Panel. The sample scripts perform this step after the WIM is installed on the device. If you included the IronKey Control Panel in your WIM image, you do not need to include this as part of the script code. The sample scripts use the IronKey Control Panel Installer to install the application to the device.

- **Validate the OS Partition**—This operation is recommended because it checks the OS partition to verify that the IronKey Control Panel component has been installed. This application is required in Windows To Go. Use the IronKeyUtil application, *check_os command*.
- **Set device in deployment mode**—Once the IronKey components have been validated, this is the last required step for W500/W700 devices. Use the IronKeyUtil application, *apply_mode command*.

The sample scripts include steps to perform these operations on only W500/W700 devices. W300 devices do not require these steps as they do not use hardware encryption and do not require the IronKey Control Panel to be installed. For W300 devices, the sample scripts apply the WIM image to the device and can optionally enable BitLocker.

Step 1: Modify scripts to support your provisioning environment

Whether you are using the sample batch or PowerShell script you will have to modify some variables and settings so the script is accurate for your provisioning requirements. The steps to change the following settings will vary depending on the script you are using. If you do not change these settings, the script may not run as expected. Details on the changes required for each script are available in these sections: “Provisioning devices using PowerShell scripts” on page 23 or “Provisioning devices using batch scripts” on page 31.

The following list is a high-level overview of the required changes for both batch and PowerShell sample scripts.

- **All device types:**
 - Specify the path and filename for the WIM image to put on the device.
 - Set the password for the default Windows Administrator account that will be created in Windows when the unattend.XML file runs on first startup of the Windows operating system. The sample PowerShell script creates an Admin account by default but the batch scripts do not.
 - If you want the script to do an offline domain join, add specific drivers, apply Windows updates, and so on, you will have to add this to the script. The sample scripts include code to enable offline domain join but it is commented out by default; you will also need to set the domain variables and any other settings required for your environment to successfully run the offline domain join.
- **For W300:** Set the Microsoft BitLocker password. The sample scripts are set up by default to use BitLocker to encrypt the data on the OS partition. If you are not using it, you should modify the script to disable BitLocker functionality.
- **For W500:**
 - Set the path and file name to: 1) IronKeyUtil (ironkeyutil.exe) and 2) IronKey Control Panel Installer application.
 - Set Admin Code and Management status (to be used by IronKeyUtil to initialize the device).

Step 2: Run the script modified in Step 1

The number of devices you can provision in one cycle will depend on the number of drive letters and USB ports available in your provisioning environment.

Insert your drives to the provisioning computer, open the appropriate command prompt and run the script with Administrator privileges. The script will provision your devices and you will see output on-screen to let you know the status of the cycle. During this process, the WIM is applied to the device. If provisioning W500/W700 devices, they will be initialized and unlocked, the IronKey Control Panel will be installed to the OS partition, and the devices will be set to Deployment mode at the end of the provisioning cycle.

GETTING STARTED

General steps to include in provisioning scripts

Important: When provisioning multiple devices, devices may become too warm to handle immediately because most USB hubs do not dissipate heat well when fully loaded. IronKey recommends (1) allowing a five minute cool-down period after the provisioning cycle ends or (2) attaching USB extender cables to allow more air-flow between devices and increase the speed of heat dissipation.

The following diagram illustrates the difference between provisioning a hardware encrypted W500/W700 device and a W300.

Figure 2-1: Overview of the steps to include in IronKey Workspace provisioning scripts

W500/W700

New/recommissioned/reset device



W300

New or provisioned device



IRONKEYUTIL COMMANDS

IronKeyUtil is a command line tool lets you to perform the following operations on IronKey Workspace W500/W700 devices. These operations are a required part of provisioning a hardware encrypted device. Whether using IronKeyUtil at the command line or incorporating the commands into a provisioning script, the utility is required for W500/W700 devices only.

- **Initialize devices (init command)**—(Required) Sets the Admin Code and unlocks the operating system (OS) partition for W500 and W700 devices. This command also sets the management option for the device. Devices must be set to managed if they will be activated and controlled by an IronKey Enterprise Management System (IronKey Enterprise Service or IronKey Enterprise Server). W300 devices do not require initialization and do not support management at this time.
- **Check the OS partition (check_os command)**—(Optional) This command verifies that required IronKey Workspace components, namely the IronKey Control Panel, are installed on the device. Hardware encrypted devices (W500 and W700), will not boot properly if this component is missing from the OS partition of the device. W300 do not require the Control Panel.
- **Set device mode (apply_mode command)**—(Required) Devices can be set in two modes: Deployment mode and Configuration mode. Devices must be set to Deployment mode before the drives are distributed to their final users.

This chapter provides information about the commands and parameters available with IronKey Workspace Command Line Utility (IronKeyUtil). The following sections are command references that describe how to run IronKeyUtil at a command line. It also provides examples of how these commands are used in the sample provisioning scripts to create Windows To Go devices.

This chapter contains information about:

- IronKeyUtil command line Help (?)
- Documentation conventions
- init command
- check_os command
- apply_mode command

IRONKEYUTIL COMMAND LINE HELP (?)

When running IronKeyUtil at a command line, the on-screen Help will list all commands and parameters available with the utility as well as the syntax to use.

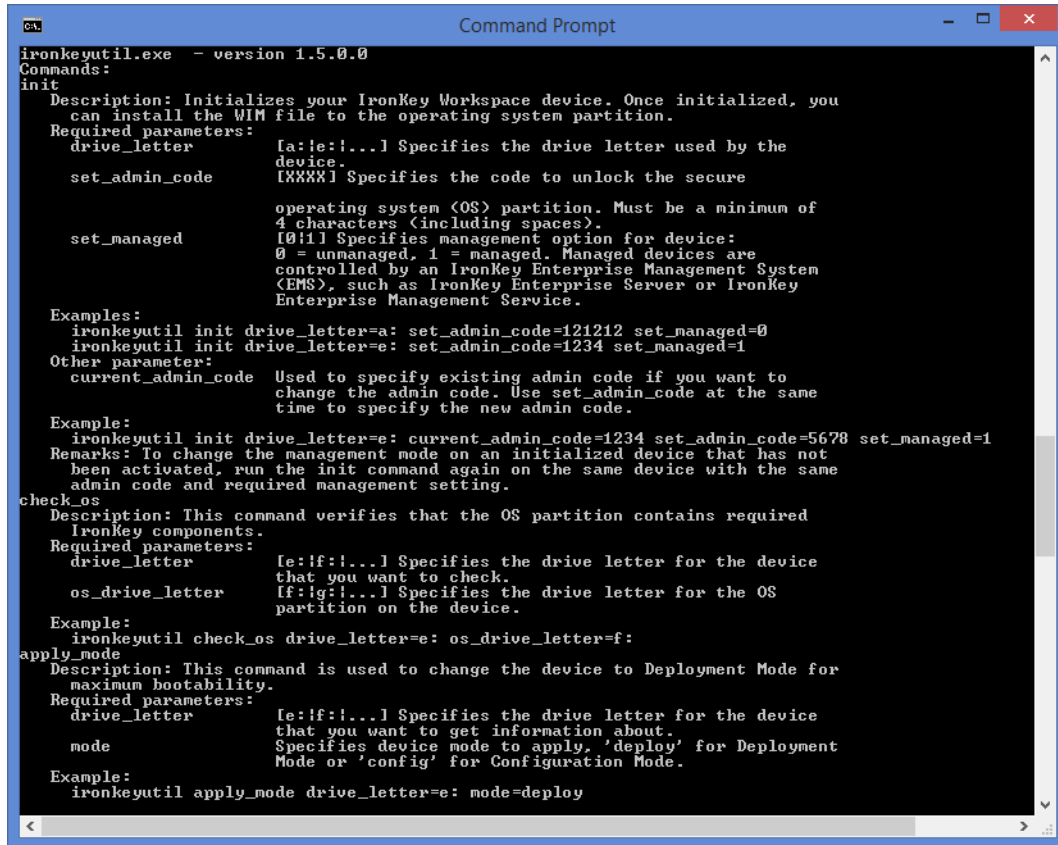
To view the on-screen Help

1. Open a Windows command prompt and change to the directory where ironkeyutil.exe is located

- Type `/?` at the command prompt.

```
c:\IronKeyUtil>ironkeyutil.exe /?
```

The Help text will display on-screen.



```

ironkeyutil.exe - version 1.5.0.0
Commands:
init
  Description: Initializes your IronKey Workspace device. Once initialized, you
  can install the WIM file to the operating system partition.
  Required parameters:
    drive_letter      [a-z:~!...~] Specifies the drive letter used by the
                      device.
    set_admin_code    [XXXXXX] Specifies the code to unlock the secure
                      operating system (OS) partition. Must be a minimum of
                      4 characters (including spaces).
    set_managed       [0|1] Specifies management option for device:
                      0 = unmanaged, 1 = managed. Managed devices are
                      controlled by an IronKey Enterprise Management System
                      (EMS), such as IronKey Enterprise Server or IronKey
                      Enterprise Management Service.
  Examples:
    ironkeyutil init drive_letter=a: set_admin_code=121212 set_managed=0
    ironkeyutil init drive_letter=e: set_admin_code=1234 set_managed=1
  Other parameter:
    current_admin_code Used to specify existing admin code if you want to
                      change the admin code. Use set_admin_code at the same
                      time to specify the new admin code.
  Example:
    ironkeyutil init drive_letter=e: current_admin_code=1234 set_admin_code=5678 set_managed=1
  Remarks: To change the management mode on an initialized device that has not
  been activated, run the init command again on the same device with the same
  admin code and required management setting.
check_os
  Description: This command verifies that the OS partition contains required
  IronKey components.
  Required parameters:
    drive_letter      [e-z:f:~!...~] Specifies the drive letter for the device
                      that you want to check.
    os_drive_letter   [f:~g:~!...~] Specifies the drive letter for the OS
                      partition on the device.
  Example:
    ironkeyutil check_os drive_letter=e: os_drive_letter=f:
apply_mode
  Description: This command is used to change the device to Deployment Mode for
  maximum bootability.
  Required parameters:
    drive_letter      [e-z:f:~!...~] Specifies the drive letter for the device
                      that you want to get information about.
    mode              Specifies device mode to apply. 'deploy' for Deployment
                      Mode or 'config' for Configuration Mode.
  Example:
    ironkeyutil apply_mode drive_letter=e: mode=deploy
  
```

DOCUMENTATION CONVENTIONS

The command sections in this chapter use the format described below.

Table 3-1: Documentation conventions

Section	Description
Name	This is the name of the IronKeyUtil command.
Syntax	<p>This section describes the syntax for the command.</p> <p>[] Square brackets</p> <p>The parameter is required in the command.</p> <p>{ } Curly brackets</p> <p>The argument must be included with the parameter.</p> <p> Separator</p> <p>The parameter must specify only one of the arguments separated by this character.</p> <p>italics</p> <p>Text that appears with <i>italic</i> formatting is sample text only and must be replaced with actual characters that you want to use for the parameter that is being set.</p>
Description	This section describes in greater detail what the command does and when to use it.
Parameters	This section describes the required and optional parameters to use with the specified command.
Examples	This section gives examples of how to use the command and provides explanations about returned values or outcomes. It also gives examples of how these commands are used in the sample provisioning scripts.

INIT COMMAND

Initializes a W500 or W700 device. Does not apply to W300 devices.

Syntax

```
c:\IronKeyUtil>ironkeyutil.exe init [drive_letter={e:}] [set_admin_code={123123}]
[set_managed={0|1}]
```

Description

The `init` command unlocks the operating system partition on hardware encrypted devices (W500 and W700) in preparation for provisioning the device with a Windows Enterprise WIM image file. The `init` command parameters require you to:

- **Set the Admin Code on the device**—The Admin Code is replaced by the device password when a W500/W700 is activated
- **Specify the drive letter used by the OS partition**
- **Specify the management status of the device**—Devices can be initialized as unmanaged or managed by an IronKey Enterprise Management System (EMS).

Devices can be initialized if they are in one of the following states:

- *New* (from factory)
- *Recommissioned* for reuse by EMS (managed W500/W700 devices). To recommission a managed device, see the EMS Admin guide.
- *Reset* to a factory state (unmanaged W500/W700 devices). You can reset an unmanaged device by exceeding the maximum 10 login attempts. *Reset* is available only if password settings in IronKey Control Panel are set to *Reset instead of self-destructing*. *Reset* is the default setting for unmanaged devices. **Important:** Disabling the *Reset* setting could permanently destroy a device if 10 login attempts are exceeded. When a device self-destructs it cannot be reused and the data is permanently inaccessible.

If your device is not in one of these states, for example, it is managed and activated with EMS, you will receive an error when you run the `init` command to initialize the drive. If the device is unmanaged, you will receive an error if the password has been changed from the default Admin Code and the device has not been Reset.

Parameters

Required parameters

set_admin_code={xxxxxxx}

This parameter sets the Admin Code on the device.

The Admin Code can be any alpha-numeric combination of characters. It is intended for Admin use only. The code unlocks the operating system partition on the device so that you can provision it with Windows To Go. It is the initial device password.

For managed devices, the Admin Code is replaced when the user sets a device password during device activation. The Admin Code set on the device must be the same as the Admin Code specified in the user account in the IronKey Enterprise Management System (EMS). For unmanaged devices, the Admin Code is the device password. Admins should change it before giving the device to the final user as a security precaution.

drive_letter={e:f:|...}

Specifies which drive letter is used by the device. You can determine which drive letter the device is using when you insert the device. When defining this parameter in a script, you can create a variable that will pass the drive letter for each device. The sample scripts use variables to set the drive letter. See the examples below for details.

set_managed={0 | 1}

Determines whether the device will be managed by EMS (IronKey Enterprise Server or IronKey Enterprise Service).

0 = unmanaged, 1 = managed

Optional parameters

current_admin_code={xxxxxxx}

Specifies the current Admin Code on an initialized device so that you can set a new Admin Code. Use this parameter with `set_admin_code` to change the Admin Code on an initialized device. If not specified, the command will use the value specified in the `set_admin_code` parameter

Examples

Example 1: Initializing a new device

This example uses the `init` command to initialize a new device with the following parameters and arguments: The device is using drive letter "e:", the Admin Code will be 123123 and the device will *not* be managed by EMS.

```
c:\IronKeyUtil>ironkeyutil.exe init drive_letter=e: set_admin_code=123123 set_managed=0
```

Results from log file: Exit Status= 0. Command `init` successfully initialized a new device.

```
-- IronKeyUtil - version 1.5.0.0 -- Log - 2015 04 04 12:11
12:11:01 - init drive_letter=e: set_admin_code=xxxx set_managed=0
12:11:01 - [E:] Command : init
12:11:40 - [E:] Exit code : 0
```

When using this code in a provisioning script, you can initialize multiple devices at once. Below is a code sample from the sample batch provisioning script that uses the `init` command to initialize devices before provisioning them with Windows To Go. The variable "%i" determines the drive letters used by all IronKey Workspace drives being provisioned in the script.

Example: `init` command used in sample batch script

```
REM THE FOLLOWING CALLS THE IRONKEY WORKSPACE COMMAND LINE UTILITY TO
REM INITIALIZE THE DEVICE AS AN UNMANAGED DEVICE, SETTING THE
REM ADMIN CODE TO "123123"
ironkeyutil.exe init drive_letter=%i set_admin_code=123123 set_managed=0
echo.
if not ERRORLEVEL 0 (
    echo ERROR: Failed to initialize %i
```

The following code from the sample PowerShell script copies the `ironkeyutil.exe` file to a folder for the disk being provisioned and sets a variable to reference where the `ironkeyutil.exe` file is now located. The script then calls an IronKey function (*Invoke-IronKeyInitializeDevice*) from the `IronKeyFunctions.psm1` module that calls `ironkeyutil.exe` and initializes the device. The script uses variables to determine the drive letter, Admin Code and Management setting: `$deviceLetter`, `$adminCode`, `$managedDevice`.

Example: PowerShell code to initialize the device

```
# BEGIN W500/W700 device preparation steps
if ($deviceType -match 'W500'){

    #This copies the ironkeyutil.exe payload to the folder for this disk
    Copy-IKPayload

    #This gets the current disk's drive letter
    $deviceLetter = Get-IronKeyDeviceLetter -disk $disk

    #This sets a variable to reference where the ironkeyutil.exe payload is located for this disk
    $ikutilFolderPath = "$RunningFromFolder\$computername\IKPayload"

    #This initializes the W500/W700 device: this will expose the encrypted OS partition
    #NOTE: This calls the ironkeyutil.exe utility to initialize the device
    Invoke-IronKeyInitializeDevice -ikutilFolderPath $ikutilFolderPath -deviceLetter $deviceLetter -
adminCode $adminCode -managedDevice $managedDevice
```

The following PowerShell code calls a function (*Invoke-IronKeyInitializeDevice*) from the sample *IronKeyFunctions.psm1* module that runs the *init* command with *ironkeyutil.exe*.

```
Function IronKey-InitializeDevice {  
    param ($deviceLetter,  
          $adminCode,  
          $managedDevice)  
    &"$RunningFromFolder\$computername\IKPayload\ironkeyutil.exe" "init"  
    "drive_letter=$deviceLetter" "set_admin_code=$adminCode" "set_managed=$managedDevice"
```

Example 2: Re-initializing a device to change the Admin Code

This example re-initializes an unmanaged device to change the current Admin Code from “123123” to “111111”.

```
c:\IronKeyUtil>ironkeyutil.exe init drive_letter=e: current_admin_code=123123  
set_admin_code=111111 set_managed=0
```

Results from log file: Exit status = 0. Command *init* was successful in re-initializing the device to change the *current_admin_code=123123* to *set_admin_code=111111*

```
-- IronKeyUtil - version 1.5.0.0 -- Log - 2015 04 04 12:24  
12:24:41 - init drive_letter=e: current_admin_code=xxxx set_admin_code=xxxx set_managed=0  
12:24:41 - [E:] Command : init  
12:25:17 - [E:] Exit code : 0
```

Example 3: Re-initializing a device to change the management status

This example re-initializes a device to change the management status from “unmanaged” to “managed”. The *set_admin_code* parameter must be the same one used when the device was first initialized.

```
c:\IronKeyUtil>ironkeyutil.exe init drive_letter=e: set_admin_code=111111 set_managed=1
```

Results from log file: Exit status = 0. Command *init* was successful in re-initializing the device to change the management status from 0 (unmanaged) to 1 (managed by EMS).

```
-- IronKeyUtil - version 1.5.0.0 -- Log - 2015 04 04 12:26  
12:26:45 - init drive_letter=e: set_admin_code=xxxx set_managed=1  
12:26:45 - [E:] Command : init  
12:27:19 - [E:] Exit code : 0
```

CHECK_OS COMMAND

Validates that the OS partition on W500/W700 devices contains the required IronKey components. This command is optional. This does not apply to W300 devices as they do not use IronKey Control Panel.

Syntax

```
c:\IronKeyUtil>ironkeyutil.exe os_check [drive_letter={e:/f:/...}]  
[os_drive_letter={f:/h:/...}]
```

Description

This command will validate that the OS partition contains the required IronKey component: IronKey Control Panel. To run the command successfully, the device must be unlocked and the OS partition mounted. In the Sample scripts provided with IronKeyUtil (W500/W700 only), this command is executed after a device has been initialized and both Windows To Go and the IronKey Control Panel have been loaded on the device. See Sample scripts for an example of how to include this validation command in a scripted provisioning task.

During validation, if the components are not found on the OS partition, an Exit code will indicate that the components are not there.

Note: When provisioning with scripts, make sure that the script includes the code to load the IronKey Control Panel on the device before it runs the `check_os` command.

Required parameters

drive_letter={e:|f:|...}

Specifies the drive letter of the read-only application partition on the device to be validated.

os_drive_letter={g:|h:|...}

Specifies the drive letter of the OS partition on the device to be validated.

Examples

This example runs the `check_os` command at the command line on a device that has already been initialized, provisioned with Windows To Go and has the IronKey Control Panel installed on the OS partition of the device.

```
c:\IronKeyUtil>ironkeyutil.exe check_os drive_letter=e: os_drive_letter=f:
```

Result in the log file: Exit code = 0. The `check_os` command found the required IronKey Control Panel application on the OS partition of the device.

```
-- IronKeyUtil - version 1.5.0.0 -- Log - 2015 04 04 14:42
14:42:53 - check_os drive_letter=e: os_drive_letter=f:
14:42:53 - [E:] Command : check_os
14:44:48 - [E:] Exit code : 0
```

You can incorporate this command as part of your batch or PowerShell provisioning script. The following code is from the sample batch script for provisioning a W500/W700 device. The script uses the command to validate that all devices currently being provisioned include the required IronKey components.

Example: Batch script code from `provisionW500-W700.cmd` to validate OS partition

```
REM THE FOLLOWING VALIDATES THE OS PARTITION TO ENSURE ALL REQUIRED COMPONENTS ARE INSTALLED
if !pubsernum! == !secsernum! (
    echo Found OS partition on %%b with serial number: !secsernum!

    ironkeyutil.exe check_os drive_letter=%a os_drive_letter=%b
    if not ERRORLEVEL 0 (
        echo.
        echo.
        echo ERROR: Check OS failed on %%b for device serial number !secsernum!
        set failedOsCheck=1
    )
)
```

Tip: See also, “Part 4: Validate the OS partition for IronKey Workspace components” on page 36.

The following PowerShell script calls a function (*Invoke-IronKeyValidateOSPartition*) from the sample *IronKeyFunctions.psm1* module that validates the OS partition for IronKey components.

Example: PowerShell script code from provisionW500-W700.cmd to validate OS partition

```
#This validates that all IronKey Control Panel components are installed to the OS Partition
$retries = 1..5
ForEach ($retry in $retries){
    Sleep -Seconds 5
    Invoke-IronKeyValidateOSPartition -ikutilFolderPath $IKPayload -deviceLetter
    $pubDiskDeviceLetter -OSDrivePath $secDiskDeviceLetter
    $result = $LASTEXITCODE
    if ($result -eq 0) {
        break
    } elseif ($retry -eq 5) {
        Write-Output "ERROR: Failed to detect IronKey Control Panel installed on $computername"
    }
}
```

The following is code from the *IronKeyFunctions.psm1* module that defines the *Invoke-IronKeyValidateOSPartition* function to run the *check_os* command with *ironkeyutil.exe* to validate the OS partition.

```
Function Invoke-IronKeyValidateOSPartition {
    param ($ikutilFolderPath,
        $deviceLetter,
        $OSDrivePath)
    &"$ikutilFolderPath\ironkeyutil.exe" "check_os" "drive_letter=$deviceLetter"
    "os_drive_letter=$OSDrivePath"
}
```

APPLY_MODE COMMAND

Allows you to set the device mode to either Deployment mode or Configuration mode.

Syntax

```
c:\IronKeyUtil>ironkeyutil.exe apply_mode [drive_letter={e:/f:/...}]
[mode={deploy/config/...}]
```

Description

Applying the device mode is the last step in provisioning a W500 or W700 device. During provisioning, the *init* command automatically puts a device in Configuration mode. Once provisioning and validating the OS partition is complete, devices should be set to Deployment mode. This mode facilitates booting the device on the widest range of qualified host computers.

The sample provisioning scripts automatically include code to set devices to Deployment mode at the end of the provisioning cycle.

Required parameters

drive_letter={e:/f:/...}

Specifies the drive letter of the OS partition on the device.

mode={deploy|config}

Specifies the operating mode to apply to the device.

deploy = Deployment Mode, config = Configuration Mode

Example

This example runs the `apply_mode` command at a command line on a device that has been initialized and provisioned with Windows To Go.

```
c:\IronKeyUtil>ironkeyutil.exe apply_mode drive_letter=e: mode=deploy
```

Result in the log file: Exit code = 0. The `check_os` command found the required IronKey Control Panel application on the OS partition of the device.

```
-- IronKeyUtil - version 1.5.0.0 -- Log - 2015 04 10 14:42
14:42:53 - apply_mode drive_letter=e: mode=deploy
14:42:53 - [E:] Command : apply_mode
14:44:48 - [E:] Exit code : 0
```

You can incorporate this command as part of your batch or PowerShell provisioning script. The following code is from the sample batch script for provisioning a W500/W700 device. The script uses the command to set all devices to Deployment mode in preparation for distributing them to final users.

Example: Setting deployment mode in sample batch script

```
REM THE DEVICE IS READY FOR USE, SWITCH THE DEVICE FROM CONFIGURATION MODE TO DEPLOYMENT MODE
ironkeyutil.exe apply_mode drive_letter=%a mode=deploy
if not ERRORLEVEL 0
    echo.
    echo.
    echo. ERROR: Failed to set %a device with serial number !secsernum! to deployment mode.
    set failedToSetDeployMode=1
```

The following code is from the sample PowerShell script. It calls the `Invoke-IronKeySetDeploymentMode` function in the `IronKeyFunctions.psm1` module to set the deployment mode for the device currently being provisioned.

Example: Setting deployment mode in sample PowerShell script

```
#This sets the W500/W700 device to deployment mode
#NOTE: This calls the ironkeyutil.exe utility to set the device in deployment mode
Invoke-IronKeySetDeploymentMode -ikutilFolderPath $IKPayload -deviceLetter $pubDiskDeviceLetter
$result = $LASTEXITCODE
if ($result -ne 0) {
    Write-Output "ERROR: Failed to set $computername to deployment mode"
} else {
    Write-Output "$computername has been set to deployment mode"
}
```

The following is code from the `IronKeyFunctions.psm1` module. It runs the `apply_mode` command in `ironkeyutil.exe` with the parameters for mode set to “deploy”.

```
Function Invoke-IronKeySetDeploymentMode {
    param ($ikutilFolderPath,
          $deviceLetter)
    &"$ikutilFolderPath\ironkeyutil.exe" "apply_mode" "drive_letter=$deviceLetter" "mode=deploy"
}
```

LOGGING AND ERROR CODES

When you run IronKeyUtil commands from a command line, the utility will display exit codes and error messages on-screen. A temporary log file is also created that captures all executed commands, exit codes, and outcomes. The log file location will display as command line output when an error occurs or the exit code is not zero (0). Review the log file for more information or to help troubleshoot any issues. The log file only captures IronKeyUtil operations. The log file does not track errors that are not related to IronKeyUtil commands when provisioning devices with scripts. Output and error messages related to running a sample script appears on-screen at run-time.

LOG FILE LOCATION AND CONTENT

The log file is created the first time you run IronKeyUtil, whether running the tool at a command prompt or calling it as part of a provisioning script. The file is saved to the “logs” subdirectory located in the folder with `ironkeyutil.exe` file. If the utility cannot write to this location, the log file will be saved to the Windows Temp directory. For example, `C:\Users\[name of user]\AppData\Local\Temp`, where [name of user] is the Windows user account for the host system.

If running the sample PowerShell script, the log folder and text file is located in the IKPayload folder, which is the default location with the `ironkeyutil.exe` file.

```
..\ironkeyutil1.5\sample scripts\IronKey_PowerShell_sample_scripts\IKPayload\logs
```

If running one of the sample batch scripts, the log folder is located in the main IronKey_batch_sample_scripts folder.

```
..\ironkeyutil1.5\sample scripts\IronKey_batch_sample_scripts\logs
```

Log files use the following naming convention, `ikutil_yymmdd_<ComputerName>.txt`, where [YYMMDD] is the date on which the utility was run and <ComputerName> is the name of the host system, for example, `ikutil_150422_DellLaptop.txt`. You must manually delete any generated log files no longer required.

The log file contains:

- IronKeyUtil version number
- Date and time stamp for each session logged
- A log of all commands executed, the returned values, and exit codes. Admin Code characters are replaced by “xxxx” so they do not appear in plain text in the log file.

Tip: To help with diagnostics, the drive letter of the device is specified on each output line in the log file.
Sample log file

The following sample log file indicates that the `init` command successfully initialized 3 devices.

`init` command
run on 3 devices

```

ikutil_150401_ProvComputer-W81.txt - Notepad
File Edit Format View Help
-- IronKeyUtil - version 1.5.0.0 -- Log - 2015 04 01 14:22
14:22:39 - init drive_letter=I: set_admin_code=xxxx set_managed=0
14:22:39 - [I:] Command : init
14:23:18 - [I:] Exit code : 0
-- IronKeyUtil - version 1.5.0.0 -- Log - 2015 04 01 14:23
14:23:18 - init drive_letter=K: set_admin_code=xxxx set_managed=0
14:23:18 - [K:] Command : init
14:23:58 - [K:] Exit code : 0
-- IronKeyUtil - version 1.5.0.0 -- Log - 2015 04 01 14:23
14:23:58 - init drive_letter=E: set_admin_code=xxxx set_managed=0
14:23:58 - [E:] Command : init
14:24:37 - [E:] Exit code : 0

```

TROUBLESHOOTING ERROR MESSAGES

Use the following table as a troubleshooting resource to look up exit codes that may appear in the log file or as command line output when running IronKeyUtil. It includes a brief description of the event and in some cases, next steps for further troubleshooting error messages.

Table 4-1: Exit codes from IronKeyUtil

Exit Code	Description
0	Success.
-1	Error including command line syntax error. See the log file for more details.
-2	Incorrect password specified.
-3	Error getting device ID. Possibly unsupported IronKey model or device communication error.
-4	Device password retry count is less than the threshold. You must use IronKey Control Panel on the device to unlock the drive. This will reset the password retry count. Once reset, you can re-run the IronKeyUtil command.
-5	You must unplug and replug the device before proceeding because of previous incorrect password attempts.
-6	The command cannot be used at this time because the device has been activated. Device will have to be recommissioned first. See the Admin Guide for your EMS product for information about how to recommission a device.
-7	The command is not supported for this IronKey model.
-8	No IronKey device found in the specified drive. Check the drive letter.

Table 4-1: Exit codes from IronKeyUtil

Exit Code	Description
-9	<i>The device is locked and the command requires that the device is unlocked.</i>
-10	<i>OS Check reports that not all components were found on the OS partition.</i> W500/W700 devices require the IronKey Control Panel to be installed on the OS partition. Make sure that you have installed the application on the device before running the <code>check_os</code> command.
-11	<i>OS Check - OS partition drive is not ready (Windows cannot read the drive.)</i>
-12	<i>OS Check - OS partition drive path is not found. Check the OS drive letter specified.</i>
-13	<i>OS Check - OS partition volume cannot be accessed. Check the OS drive letter specified.</i>
-14	<i>Unable to get device firmware version information.</i>
-15	<i>Unable to read device version information, open error.</i>
-16	<i>Unable to read device version information, read error.</i>
-17	<i>Device does not have minimum required firmware version for this command</i> Update your device to the version supported with this release. See “Supported devices” on page 5. Device updates are available through EMS for managed devices and on the <i>IronKey Support site</i> for unmanaged devices.
-18	<i>Device software not supported by this version of the utility.</i> Update your device to the version supported with this release. See “Supported devices” on page 5. Device updates are available through EMS for managed devices and on the <i>IronKey Support site</i> for unmanaged devices.
-20	<i>OS drive letter serial number does not match device drive letter serial number</i>
-21	<i>OS drive letter specified is not an IronKey drive: {drive letter}</i>
-22	<i>Cannot get device information from device at OS drive letter</i>
-24	<i>OS drive letter specified does not match device drive letter specified</i>
-25	<i>This utility is not supported while booted in Windows To Go.</i>

PROVISIONING DEVICES USING POWERSHELL SCRIPTS

The sample PowerShell script, when configured for your provisioning environment, will provision IronKey Workspace W300, W500, and W700 devices. The script is meant as a starting point from which you can create a custom script that meets your company's requirements. The script is located in the `ironkeyutil1.5\sample_scripts\IronKey_PowerShell_sample_scripts.zip` file.

Note: This chapter was written for administrators who will be using PowerShell scripts to provision IronKey Workspace devices. It assumes that the reader has a general understanding of PowerShell and how to create and run PowerShell scripts.

REQUIREMENTS

To provision devices using the sample PowerShell script you will need the following items:

- Provisioning computer running Windows 8.1 Enterprise. The sample PowerShell script must be run in Administrator mode in Windows PowerShell. The sample scripts will work with Windows 10 Enterprise Preview, however, you must remove the section that verifies that the host system is running Windows 8.1 Enterprise.
- USB powered hubs if provisioning multiple devices
- PowerShell script package (`IronKey_PowerShell_sample_scripts.zip`)—Extract the script package and make sure the provisioning computer can access the scripts and supporting folders. It is recommended that you leave the file structure as-is because the sample script uses this file structure.
- IronKey Workspace devices—W500/W700 devices must be at version 4.3, see “Supported devices” on page 5.

WHAT'S IN THE ZIP FILE?

The extracted `IronKey_PowerShell_sample_scripts` folder, contains the PowerShell script to provision devices and supporting files and folders (including IronKeyUtil and IronKey Control Panel).

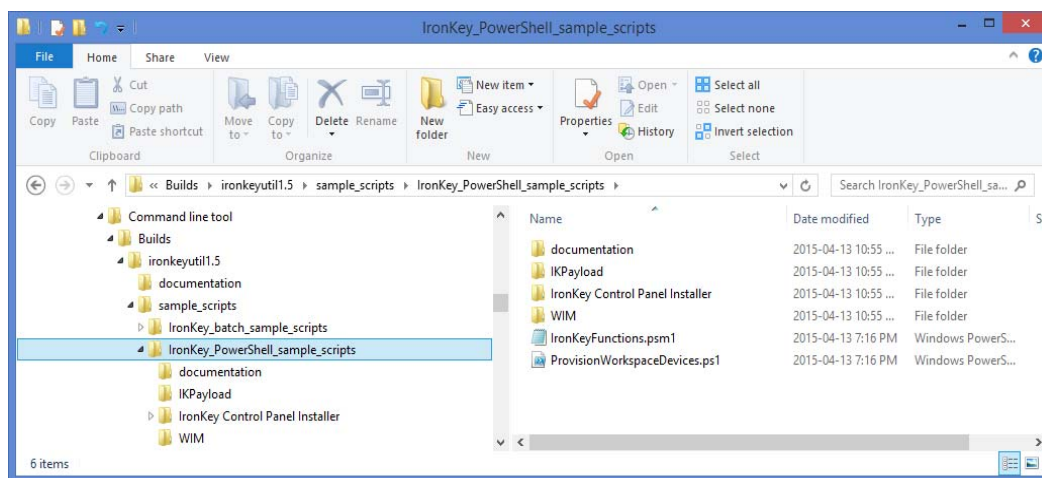


Table 5-1: Contents of sample PowerShell script folder

Filename	Description
<code>documentation\IronKeyUtil_EULA.HTML</code>	The End User License Agreement
<code>documentation\IronKeyUtil_AdminGuide.pdf</code>	The Admin Guide that you are currently reading.
IKPayload	<p>This folder contains:</p> <ul style="list-style-type: none"> <code>ironkeyutil.exe</code>—The executable file for IronKey Workspace Command Line Utility. IronKeyUtil is required to initialize W500/W700 devices during the provisioning process and to validate the OS partition to ensure that IronKey components are installed. DLL files—These files are dependencies that are required to run <code>ironkeyutil.exe</code>.
IronKey Control Panel Installer folder	<p>Required for W500/W700 devices only. This folder contains the executable file and supporting files that are required to install the IronKey Control Panel to the OS partition. The application must be available on W500/W700 devices when booted in Windows To Go. This is especially important for managed devices because the application communicates with EMS to receive device policy and update notifications.</p> <p>The sample script will install this application on the device during the provisioning process.</p> <p>An instance of the IronKey Control Panel is already installed on the application partition from factory; it is accessible when the device is not booted in Windows To Go.</p>

Table 5-1: Contents of sample PowerShell script folder

Filename	Description
WIM	This folder should contain the WIM image file (if using the sample script) that you want to install on your devices during provisioning. You will need to update the sample PowerShell script so the \$WIMFile variable uses your WIM filename; the default filename in the script is <code>install.wim</code> .
IronKeyFunctions.psm1	This is a PowerShell module that contains important IronKey functions that are used in the <code>ProvisionWorkspaceDevices.ps1</code> script during provisioning. These functions are required to perform specific operations on the device during provisioning such as initializing a W500/W700 device and installing IronKey Control Panel.
ProvisionWorkspaceDevices.ps1	<p>This is the sample PowerShell script that will provision IronKey Workspace devices.</p> <p>Important: You must modify some variables in this file to include settings that are specific to your provisioning environment, for example to set the WIM filename (\$WIMFile) or to set the Admin Code (\$adminCode) and management status (\$managedDevice) for W500/W700 devices.</p> <p>The original sample script was provided by James Bannan. Imation adapted it to show how to use the IronKey Workspace Command Line Utility (<code>ironkeyutil.exe</code>) to provision IronKey Workspace W500 and W700 devices using scripts. The script also supports W300 devices.</p> <p>This script uses functions defined in the <code>IronKeyFunctions.psm1</code> PowerShell module.</p>

SAMPLE POWERSHELL SCRIPT

The sample script will provision W300, W500, or W700 devices with Windows To Go. You can provision a mix of device types in the same provisioning cycle. The script includes comments that describe what each section is attempting to do. Although the script has many common elements that apply to all device types, some elements are specific to either W300 or W500/W700 devices. The script calls some specific IronKey functions from the IronKey Functions module (`IronKeyFunctions.psm1`).

W300 devices

The script enables BitLocker drive encryption when provisioning devices. You must set the BitLocker password before running the script. By default, the script will not prompt you to enter this password at run-time. This is because some provisioning environments require that devices are provisioned without administrator intervention. If required, you can change this variable to force the script to prompt for the password.

W500/W700 devices

Hardware encrypted W500/W700 devices require the following extra steps during provisioning:

- Initialize and unlock the encrypted OS partition using the IronKey Workspace Command Line Utility
- Install IronKey Control Panel to the device so that it will be available when booted in Windows To Go
- Validate that all IronKey Control Panel components are installed on the drive
- Set the device to deployment mode. This is the final step in provisioning these devices

Common components

In addition to setting up the drives and installing the WIM image, the script will also do the following:

- Create a Windows unattend.xml file
- Create an Administrator account in Windows—You must set the variable
- Perform Offline Domain Join—To perform an offline domain join, you must meet the requirements set by Microsoft, including ensuring that the provisioning computer is a member of the domain and the user has privileges to join computers to the domain. The script includes commands that call the Microsoft djoin.exe tool to perform a offline domain join. For more information about offline domain joining, see, See also *DirectAccess in Windows Server*: <https://technet.microsoft.com/en-us/library/dn636118.aspx>

<https://technet.microsoft.com/en-us/library/offline-domain-join-djoin-step-by-step%28v=ws.10%29.aspx>

- Add device to DirectAccess Group—If your configuration is not set up for DirectAccess, the script will skip this step.

BEFORE YOU RUN THE SCRIPT

Make sure that you extract the batch script folder (`IronKey_PowerShell_sample_scripts.zip`) from the main `ironkeyutil1.5.zip` package. The sample script `ProvisionWorkspaceDevices.ps1` is meant as a starting point from which you can create a custom script that meets your provisioning requirements.

As a minimum, you must modify the following variables in the variable section at the beginning of the script.

- **\$WIMFile**—(required) Specify the WIM filename. The sample scripts expect the WIM to be located in the WIM folder. If you change this location, you must specify the path as well as the filename.
- **\$WindowsAdminPassword**—Change the password from the default value, “Welcome123”
- **\$BitlockerPassword**—(W300 only) Change the password from the default value, “Welcome123”

For W500 devices:

- **\$adminCode**—Set the Admin Code (required for IronKeyUtil to initialize W500/W700 devices)
- **\$managedDevice**—Set the Management Status devices (required for IronKeyUtil to initialize W500/W700 devices)
- **\$IKPayload**—Sets the path to `ironkeyutil.exe`. If you changed the location from the default structure in the sample scripts folder, you must specify the new path.
- **\$CPPayload**—Sets the path to the IronKey Control Panel Installer folder. If you changed the location from the default structure in the sample scripts folder, you must specify the new path.
- Optional—Define any other variables in this section as appropriate for your provisioning environment, for example, the image index for the WIM and other image related details, such as Domain, Time Zone and so on.

Figure 5-2: Variable section of the ProvisionWorkspaceDevices.ps1 script

```

ProvisionWorkspaceDevices.ps1 X
49
50 <#####
51 TODO: Set the following variables to match the needs of your provisioning environment
52 #####>
53
54 #This specifies if the script is being run via SCCM ($True) or directly from a PowerShell prompt on the host computer ($False)
55 $usingSCCM = $True
56 if ($usingSCCM -eq $True) {
57     $BitLockerPassword = ConvertTo-SecureString -AsPlainText -Force -String "Welcome123" # TODO: Set the BitLocker password
58     $WindowsAdminPassword = "Welcome123" # TODO: Set the local WTG Administrator account password
59 }
60
61
62 #This sets the WIM to be used for provisioning
63 $WIMFile = "$RunningFromFolder\WIM\install.wim"
64
65 #This sets the image index in the WIM to be used
66 $Index = '1'
67
68 #This sets the folder to the ironkeyutil.exe
69 $IKPayload = "$RunningFromFolder\IKPayload"
70
71 #This sets the folder to the IronKey Control Panel
72 $CPLPayload = "$RunningFromFolder\IronKey Control Panel Installer"
73
74 #This sets the Workspace device's Admin Code as part of device initialization
75 $adminCode = "123123"
76
77 #This sets the management status of the Workspace as managed(1) or unmanaged(0)
78 $managedDevice = 0
79
80 #The following set variables for use in the unattend XML file
81 $InputLocale = 'en-US'
82 $SystemLocale = 'en-US'
83 $UserLocale = 'en-US'
84 $UILanguage = 'en-US'
85 $OrgName = 'YourOrgName'
86 $TimeZoneName = 'U.S. Eastern Standard Time'
87
88 #The following set variables for use in domain joining
89 $DomainName = 'YourDomainName'
90 $OUPath = 'OU=WTGGroup,OU=YourOrgName,DC=YourOrgName,DC=com'
91
92 #####
  
```

These variables are referenced in IronKey functions located in the IronKeyFunctions.psm1 module. The provisioning script call these functions when performing specific operations during a provisioning cycle. For example, in Table 5-3, the function Invoke-IronKeyInitializeDevice is using the script variables \$adminCode and \$managedDevice. When the provisioning script runs this function (see Table 5-4), all W500/W700 devices currently being provisioned will be initialized with the settings defined in these variables.

Table 5-3: Invoke-IronKeyInitializeDevice function in IronKeyFunctions.psm1 file

```

Function Invoke-IronKeyInitializeDevice {
    param ($ikutilFolderPath,
           $deviceLetter,
           $adminCode,
           $managedDevice)
    "&"$ikutilFolderPath\ironkeyutil.exe" "init" "drive_letter=$deviceLetter"
    "set_admin_code=$adminCode" "set_managed=$managedDevice"
}
  
```

Table 5-4: Code in ProvisionWorkspaceDevices.ps1 script that calls the Invoke-IronKeyInitializeDevice function

```

#This initializes the W500/W700 device: this will expose the encrypted OS partition
#NOTE: This calls the ironkeyutil.exe utility to initialize the device
Invoke-IronKeyInitializeDevice -ikutilFolderPath $ikutilFolderPath -deviceLetter $deviceLetter -
adminCode $adminCode -managedDevice $managedDevice
  
```

For more information about other IronKey functions used in the script, see

To customize the sample provisioning script

1. Open ProvisionWorkspaceDevices.ps1 in a text editor or Windows PowerShell ISE.
2. Locate the variable section, near the beginning of the script, and modify the following variables:

- **\$BitlockerPassword** and **\$WTGAdminPassword**—Replace the default password “Welcome123” with a new password for these variables.

```
#This specifies if the script is being run via SCCM ($True) or directly from a PowerShell
prompt on the host computer ($False)
$usingSCCM = $True
if ($usingSCCM -eq $True) {
    #NOTE: The standard BitLocker password requires the following:
    #      * at least 1 upper case character
    #      * at least 1 number
    #      * a minimum 8 characters
    $BitlockerPassword = ConvertTo-SecureString -AsPlainText -Force -String "Welcome123" #
    TODO: Set the BitLocker password

    $WindowsAdminPassword = "Welcome123" #
    TODO: Set the local WTG Administrator account password
}
```

- **\$WIMFile**—Replace install.wim filename with the name of your WIM file. If the file is not located in the WIM folder, change the folder path.

```
$WIMFile = "$RunningFromFolder\WIM\install.wim"
```

- Modify the **\$IKPayload** and **\$CPPayload** variable if you changed the default location of the **ironkeyutil.exe** file or the **IronKey Control Panel Installer** folder.

```
#This sets the folder to the ironkeyutil.exe
$IKPayload = "$RunningFromFolder\IKPayload"

#This sets the folder to the IronKey Control Panel
$CPPayload = "$RunningFromFolder\IronKey Control Panel Installer"
```

- **\$adminCode**—Replace 123123 with the code you want to set on devices

```
$adminCode = "123123"
```

- **\$managedDevice**—Set the management status to 0 for unmanaged devices or 1 for managed devices

```
$manageDevice = 1
```

3. Modify any other variables in this section according to your requirements. For example, if you want the script to do an offline domain join, set the **\$DomainName** and **\$OUPath** variables.

```
#The following set variables for use in domain joining
$DomainName = 'YourDomainName'
$OUPath = 'OU=WTGGroup,OU=YourOrgName,DC=YourOrgName,DC=com'
```

4. Save the script and close the editor.

RUNNING THE POWERSHELL SCRIPT

You must run the PowerShell script with Administrative privileges. Review the section “Before you run the script” and modify the script to use settings specific to your configuration.

You can provision W300, W500, and W700 devices in the same cycle. The number of drive letters and ports available will determine how many devices you can provision in one cycle. The script will provision devices in parallel. It creates an output folder for each device with the device type and serial number as the folder name. When provisioning W500/W700 devices, IronKeyUtil will capture all commands and output for each device and save them in the respective device folder in the logs directory.

To run the PowerShell script

1. Log on to the provisioning computer (running Windows 8.1 Enterprise).

2. Insert IronKey Workspace devices in the USB ports or USB hubs (USB 3.0 ports are recommended) connected to the provisioning computer. If you are using USB 2.0 ports, provisioning may take longer.
3. Verify that the drives are mounted using Windows Disk Management before starting the provisioning script.
4. Run Windows PowerShell as Administrator. You can also run a Windows command prompt as Administrator; see Step 6 for command syntax to run the script.
5. At the command prompt, change the directory to the *IronKey_PowerShell_sample_scripts* folder (where the *ProvisionWorkspaceDevices.ps1* script is located).

```
.\ironkeyutil1.5\sample_scripts\Ironkey_PowerShell_sample_scripts>
```

6. Type **ProvisionWorkspaceDevices.ps1** to run the script.

If you are running the script from a Windows command prompt (with Administrator privileges), type the following command syntax: **powershell.exe -file ProvisionWorkspaceDevices.ps1**

During the provisioning process, File Explorer windows may open during the provisioning cycle and output to the command line will indicate what is happening at various stages of the process.

7. When the cycle has finished, the output for all jobs will display on-screen. The message,

```
Provisioned <x> devices
```

indicates the number of devices provisioned in this cycle. Review the script output for errors or issues. You can also review the IronKeyUtil log file (*ikutil_YYMMDD_<computer name>.txt*) to see all operations that called the IronKeyUtil application.

8. After the cycle has finished, it is recommended that you allow a five minute cool-down period before removing devices. The provisioning process can warm devices to temperatures that may make them uncomfortable to handle immediately.
9. Review the command line output and Remove the provisioned devices and complete any final setup procedures required. If you want to start a new provisioning cycle, insert new devices and repeat this procedure.

Sample script output

The following code shows a sample of the on-screen output that will display at the command prompt during a provisioning cycle. The script created a device name using WTG and the serial number of the device. The device being provisioned in the output below is "WTG-02205945".

The section “-----WTG-02205945 START-----” is deploying the WIM image to the device. Following this section, the output indicates that the IronKey Control Panel is being installed to the device, the script has verified that the components are installed, and the device mode is being set to Deployment. The final line of the script indicates that 1 device has been provisioned in this cycle.

```
This is a sample script provided by Imation.

**WARNING** ActiveDirectory is not available; it will not be loaded.

    Directory:
E:\software\builds\ironkeyutil1.5_b53\sample_scripts\IronKey_PowerShell_sample_scripts

Mode                LastWriteTime         Length Name
----                -
d-----          4/20/2015   4:41 PM           WTG-02205945
Starting Deployment of WTG-02205945

Provisioning 1 devices, please wait...

-----= WTG-02205945 START =-----
WTG-02205945 >> MSFT_Volume (ObjectId = "\\?\Volume{3bd574f1-e79d-11e4-82e1-e0db...})
WTG-02205945 >> MSFT_Volume (ObjectId = "\\?\Volume{3bd574f8-e79d-11e4-82e1-e0db...})
WTG-02205945 >>
WTG-02205945 >> Deployment Image Servicing and Management tool
WTG-02205945 >> Version: 6.3.9600.17031
WTG-02205945 >>
WTG-02205945 >> Applying image
WTG-02205945 >> The operation completed successfully.
WTG-02205945 >> Boot files successfully created.
WTG-02205945 >> H:\Windows\setup\scripts
WTG-02205945 >> **WARNING** ActiveDirectory module is not available; skipping call to Add-
ADPrincipalGroupMembership
-----= WTG-02205945 END =-----

Installing the IronKey Control Panel on WTG-02205945
Successfully installed IronKey Control Panel on WTG-02205945
Device mode change takes effect the next time the device is plugged in. You must remove and re-plug
the device before running other commands on this device.

WTG-02205945 has been set to deployment mode

Ticks                : 6208280005
Days                 : 0
Hours                : 0
Milliseconds         : 828
Minutes              : 10
Seconds              : 20
TotalDays            : 0.0071855092650463
TotalHours           : 0.172452222361111
TotalMilliseconds    : 620828.0005
TotalMinutes         : 10.3471333416667
TotalSeconds         : 620.8280005

Provisioned 1 devices
```

PROVISIONING DEVICES USING BATCH SCRIPTS

You can provision IronKey Workspace devices (W300, W500, and W700) using the sample Windows batch scripts. The scripts are located in the `provision_batch_file_samples.zip` file. This file is included as part of the IronKey Workspace Command Line Utility package (`ironkeyutil1.5.zip`). You should be familiar with running batch scripts and using command line tools and interfaces.

The sample scripts will provision multiple devices of the same type. The number of drive letters and ports available will determine how many devices you can provision in one cycle. Hardware encrypted devices (W500 and W700) can be provisioned in the same cycle using the `provisionW500-W700.cmd` script. W300 devices must be provisioned separately using the `provisionW300.cmd` script. For details about the scripts, see “Sample batch provisioning scripts” on page 33.

REQUIREMENTS

To provision devices using a batch script you will need the following items:

- Provisioning computer running Windows 8.1 Enterprise. You must run the script in Administrator mode. Other environments are not supported by the sample script and further customization will be required to run the script in a different environment. For example, to use Windows 10 Enterprise Preview, the host computer must also be running Windows 10 Enterprise and you must remove the section of the sample script that requires Windows 8.1 Enterprise.
- USB hubs to provision multiple devices in one cycle (optional).
- Access to the location of the extracted batch script folder (`provision_batch_file_samples.zip`) from the provisioning computer. This folder contains sample scripts, the IronKey Control Panel application and the IronKeyUtil command line tool. It is recommended that when you extract the file, you leave the file structure as-is; the sample script uses this file structure.
- Windows image (WIM) file to install to the OS partition of the device.
- IronKey Workspace devices—W300, or W500/W700 devices at version 4.3

Tip: For more information about general provisioning requirements, see “Requirements for provisioning devices” on page 5.

WHAT'S IN THE ZIP FILE?

The following table provides details about the files included in the `IronKey_batch_sample_scripts` folder.

Table 6-1: Contents of batch file sample script folder

Filename	Description
d.vbs	A Visual Basic script used by the sample provisioning scripts that returns a list of all W500/W700 public partitions.
ironkeyutil.exe	IronKey Workspace Command Line Utility. A command line tool that is required to initialize W500/W700 devices before provisioning with Windows To Go. This utility is also used to check the OS partition to verify that required IronKeyUtil components are installed, and to set the device in Deployment mode.
DLL files	These files are dependencies that are required to run <code>ironkeyutil.exe</code> .
IronKey-GetSerialNumber.ps1	A PowerShell Cmdlet used in the sample batch provisioning scripts to get the serial number of an IronKey Workspace drive. For example, when validating the OS partition of W500/W700 drives, the sample W500/W700 script uses this script to get the device serial number and match it with the OS partition of all devices currently being provisioned. Once matched, the provisioning script can validate that the OS partition contains the required IronKey components. See "Part 4: Validate the OS partition for IronKey Workspace components" on page 36.
wtg.ps1	<p>A PowerShell script that demonstrates how to provision one or more USB drives with Windows To Go. It also supports performing an offline domain join. This script was adapted by Imation from the original script that is available on the Microsoft TechNet Web site. For more information about the original script, see https://technet.microsoft.com/en-ca/library/jj721578.aspx</p> <p>This script has been modified from the original sample script to:</p> <ol style="list-style-type: none"> 1. Increase the default primary partition size from 350 MB to 540 MB. 2. Increase the sleep duration between partition creation and partition formatting from the default setting of 1 second to 5 seconds to give the partition time to settle before formatting.
NoDriveTypeAutoRun.reg	A registry file used by the sample provisioning scripts to temporarily disable Auto Run on the host computer when initializing the devices. The script restores the original host settings after device initialization. See "Part 1: Initialize all devices" on page 33.
provisionW300.cmd	The sample provisioning script to provision W300 devices. You can customize this script to meet your specific configuration requirements. This script does not support other device types.

Table 6-1: Contents of batch file sample script folder

Filename	Description
provisionW500-W700.cmd	The sample provisioning script to provision W500 and W700 devices. You can use both device types in the same cycle when running this script. This script does not support W300 devices.
documentation\IronKeyUtil_EULA.HTML	The End User License Agreement (EULA).
IronKey Control Panel Installer <i>folder</i>	<p>Required for W500/W700 devices only. This folder contains the executable file and supporting files that are required to install the IronKey Control Panel to the OS partition. The application must be available on W500/W700 devices when booted in Windows To Go. This is especially important for managed devices because the application communicates with EMS to receive device policy and update notifications.</p> <p>The sample script will install this application on the device during the provisioning process.</p> <p>An instance of the IronKey Control Panel is already installed on the application partition from factory; it is accessible when the device is not booted in Windows To Go.</p>

SAMPLE BATCH PROVISIONING SCRIPTS

IronKey Workspace W500/W700 script

The sample W500/W700 provisioning script must be run with Administrator privileges and is configured to run on Windows 8.1 Enterprise. The sample script is designed to provision up to 7 devices depending on the number of ports and drive letters are available in your environment. There are four main sections that complete the following provisioning tasks: Initialize device, Provision devices with WIM, Install IronKey Control Panel, Validate the OS partition for IronKey components and set device to Deployment mode.

Part 1: Initialize all devices

This section prepares devices for provisioning with the WIM file by initializing them using IronKeyUtil. The default settings in the script create an unmanaged device with a generic Admin Code. You will need to customize the `set_managed` and `set_admin_code` parameters to your specific requirements. See “To customize the W300 sample script” on page 38.

The following code from the sample script calls the `init` command from `ironkeyutil.exe` to initialize W500/W700 devices.

```
REM *****
REM PART 1 START : INITIALIZING DEVICES

echo Disabling Auto Play temporarily
echo.

REM THE FOLLOWING EXPORTS THE HOST SYSTEM'S CURRENT/DEFAULT AUTO PLAY SETTINGS
REM TO FILE; THIS SETTING IS RESTORED AT THE END OF THE SCRIPT TO RETURN
reg export HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer default.reg /y 1>NUL
2>NUL

REM THE FOLLOWING IMPORTS A CHANGE TO THE EXPLORER KEY'S "NoDriveTypeAutoRun" VALUE
REM THAT INSTRUCTS THE SYSTEM TO NOT AUTO RUN BOOTABLE DRIVES (0X000000fd)
reg import NoDriveTypeAutoRun.reg 1>NUL 2>NUL

echo Detecting and initializing devices
echo Please wait...
echo.

REM THE FOLLOWING CALLS A VBSCRIPT WHICH DETECTS DEVICES AND RETURNS
REM THE DRIVE LETTERS FOR EACH DEVICE FOUND
for /f "delims=" %i in ('cscript d.vbs //nologo') do (
    echo Found public partition: %i
    set foundPubPartition=1

    echo Configuring %i device, please wait...
    echo.
    REM THE FOLLOWING CALLS THE IRONKEY WORKSPACE COMMAND LINE UTILITY TO
    REM INITIALIZE THE DEVICE AS AN UNMANAGED DEVICE, SETTING THE
    REM ADMIN CODE TO "123123"
    ironkeyutil.exe init drive_letter=%i set_admin_code=123123 set_managed=0
    echo.
    if not ERRORLEVEL 0 (
        echo ERROR: Failed to initialize %i
    )
)

echo Re-enabling saved Auto Play settings
REM THE FOLLOWING RESTORES THE EXPORTED AUTO PLAY SETTINGS
reg import default.reg 1>NUL 2>NUL
del default.reg
echo.

if [%foundPubPartition%] == [0] (
    REM NO DEVICES WERE FOUND
    goto :DETECT_PUBLIC_FAILED
)

echo Devices are ready for image deployment
echo.
REM *****
REM PART 1 END : INITIALIZING DEVICES
REM *****
```

Part 2: Provision initialized devices with specified WIM file

The second part of the batch script runs a sample PowerShell script (`wtg.ps1`) to provision devices with the WIM image. Make sure that you specify the WIM location in the batch script. See “To customize the W300 sample script” on page 38. The WIM file should be a Windows 8.1 Enterprise image that has been generalized using sysprep and includes any company-specific files or applications that are required. If devices will be used on qualified Mac computers, consider including the Boot Camp driver package that supports Windows 8.1 on a Mac. For more information, see the *IronKey Workspace IT Administrator Handbook*.

The PowerShell script (`wtg.ps1`) is a modified version of the sample script from Microsoft that is available on the Microsoft Technet Web site at https://technet.microsoft.com/en-ca/library/jj721578.aspx#wtg_advanced_dep

Microsoft recommends that when using their sample PowerShell script, you should also configure the PowerShell execution policy from the default *Restricted* policy to the *RemoteSigned* policy using the following command `Set-ExecutionPolicy RemoteSigned`. The sample batch script provided sets the PowerShell execution policy to *RemoteSigned* so this step is not necessary.

The *RemoteSigned* execution policy allows a script that is created locally to run but will prevent the computer from running unsigned scripts from the Internet. If set to the default *Restricted* execution policy, the computer will not allow a script to run without explicit permission to do so.

The following sample code provisions devices with the WIM file.

```
REM *****
REM PART 2 START : PROVISIONING DEVICES
REM *****
echo Devices being provisioned with the following image: %mywim%
echo.
REM THE FOLLOWING EXECUTES A POWERSHELL SCRIPT WHICH PROVISIONS THE INITIALIZED
REM DEVICE'S NOW-READY OS PARTITION WITH THE SPECIFIED WIM
REM NOTE: THIS wtg.ps1 POWERSHELL SCRIPT IS AN EXAMPLE SCRIPT PROVIDED BY
REM       MICROSOFT AS A SAMPLE OF HOW TO PROVISION MULTIPLE WINDOWS TO GO
REM       DRIVES.
REM       THIS POWERSHELL SAMPLE SCRIPT IS PART OF MICROSOFT'S "Deploy Windows
REM       To Go" ARTICLE ON TECHNET:
REM       https://technet.microsoft.com/en-ca/library/jj721578.aspx
powershell Set-ExecutionPolicy RemoteSigned
powershell -File wtg.ps1 -InstallWIMPath %mywim%
if errorlevel 1 (
    echo An error occurred, please check the error information
) else (
    echo Successfully provisioned device
)
echo.
REM *****
REM PART 2 END : PROVISIONING DEVICES
REM *****
```

Part 3: Install IronKey Control Panel to devices

The sample script includes code to install the IronKey Control Panel to the OS partition of the device. IronKey Control Panel is an important application for W500/W700 devices. Out of the box, the Control Panel comes pre-installed on the read-only application partition of the device. However, you must also install it to the OS partition during provisioning so that users can access it when booted in Windows To Go, for example to change their device password or to receive device policy and update notifications (managed devices only).

Note: W300 do not require the Control Panel to be installed on the device.

The following code installs the IronKey Control Panel to the device.

```
REM *****
REM PART 3 START : INSTALL CONTROL PANEL TO DEVICES
REM *****

REM FIND THE PUBLIC PARTITION, AND THEN THE OS PARTITION
echo Detecting OS partition, please wait...
REM THE FOLLOWING DETECTS DEVICES' OS PARTITION AND RETURNS
REM THE DRIVE LETTERS FOR EACH PARTITION FOUND
for /f %j in ('wmic volume get DriveLetter^, Label ^| find "UFD-Windows"') do (
    echo Found OS partition on %j
    set foundSecPartition=1

    REM THE FOLLOWING INSTALLS THE CONTROL PANEL ON THE OS PARTITION
    echo Installing IronKey Control Panel to %j...
    pushd %cpInstallerFolder%
    %cpInstaller% /f %j
    popd
    echo The IronKey Control Panel is now installed
)

if [%foundSecPartition%] == [0] (
    echo Failed to find the OS partition
    goto :DETECT_SECURE_FAILED
)
REM *****
REM PART 3 END : INSTALL CONTROL PANEL TO DEVICES
REM *****
```

Part 4: Validate the OS partition for IronKey Workspace components

As a final step in the provisioning process, the sample script validates that all devices include the required IronKey component: IronKey Control Panel. Using IronKeyUtil (`check_os` command), the script checks the OS partition to make sure IronKey Control Panel is installed. This step is recommended but not required as a provisioning step. Once validated, the script again uses IronKeyUtil to run the `apply_mode` command to set the device in Deployment mode.

The following sample code from “Part 4: Validate OS Partition” section of the script, shows how to validate the OS partition and set the device mode to “deploy”.

```
REM THE FOLLOWING VALIDATES THE OS PARTITION TO ENSURE ALL REQUIRED COMPONENTS ARE INSTALLED
    if !pubsernum! == !secsernum! (
        echo Found OS partition on %%b with serial number: !secsernum!

        ironkeyutil.exe check_os drive_letter=%a os_drive_letter=%%b
        if not ERRORLEVEL 0 (
            echo.
            echo.
            echo ERROR: Check OS failed on %%b for device serial number !secsernum!
            set failedOsCheck=1
        ) else (
            REM THE DEVICE IS READY FOR USE, SWITCH THE DEVICE FROM CONFIGURATION MODE TO
DEPLOYMENT MODE
            ironkeyutil.exe apply_mode drive_letter=%a mode=deploy
            if not ERRORLEVEL 0 (
                echo.
                echo.
                echo ERROR: Failed to set %a device with serial number !secsernum! to deployment
mode.
                set failedToSetDeployMode=1
            )
        )
    )
```

IronKey Workspace W300 script

You must run the sample W300 script with Administrator privileges on a system running Windows 8.1 Enterprise. W300 devices do not use hardware encryption and cannot be managed by EMS. As a result, you do not have to initialize the device using IronKeyUtil, install the IronKey Control Panel, or validate the OS partition and set the device to deployment mode. Instead, the script is set up to use BitLocker drive encryption to protect data on the OS partition.

You must set the BitLocker password before running the script. By default, the script will not prompt you to enter this password at run-time. This is because some provisioning environments require that devices are provisioned without administrator intervention. If required, you can change this variable to force the script to prompt for the password.

The sample script has two main parts: Provisioning devices, and getting the device serial number.

Part 1: Provisioning devices

This section calls the `wtg.ps1` PowerShell script to apply the WIM image to the OS partition and uses the BitLocker password variable when enabling BitLocker encryption.

```
REM *****
REM PART 1 START : PROVISIONING DEVICES
REM *****
set foundPartition=0
REM THE FOLLOWING EXECUTES A POWERSHELL SCRIPT WHICH PROVISIONS THE
REM DEVICE'S OS PARTITION WITH THE SPECIFIED WIM
REM NOTE: THIS wtg.ps1 POWERSHELL SCRIPT IS AN EXAMPLE SCRIPT PROVIDED BY
REM       MICROSOFT AS A SAMPLE OF HOW TO PROVISION MULTIPLE WINDOWS TO GO
REM       DRIVES.
REM       THIS POWERSHELL SAMPLE SCRIPT IS PART OF MICROSOFT'S "Deploy Windows
REM       To Go" ARTICLE ON TECHNET:
REM       https://technet.microsoft.com/en-ca/library/jj721578.aspx
echo Devices being provisioned with the following image: %mywim%
echo.

powershell Set-ExecutionPolicy RemoteSigned
powershell -File wtg.ps1 -InstallWIMPath %mywim% -BitLockerPassword %bitLockerPassword%
if errorlevel 1 (
    echo An error occurred, please check the error information
) else (
    echo Successfully provisioned device
)
echo.
REM *****
REM PART 1 END : PROVISIONING DEVICES
REM *****
```

Part 2: Example of how to get the device serial number

This section shows you how to get the device serial number. This function is useful when managing Windows To Go devices because you can set the serial number as the computer name, for example, to use when joining a domain. The script calls a Windows PowerShell Cmdlet (`IronKey-GetSerialNumber.ps1`) that returns the serial number of the drive.

The following table shows the code from the script. It searches for an OS partition and then calls the PowerShell Cmdlet to get the serial number from the partition.

Table 6-2: ProvisionW300.cmd script - Get serial number

```

REM *****
REM PART 2 START : EXAMPLE OF HOW TO GET SERIAL NUMBER FROM DEVICE
REM *****

SetLocal EnableDelayedExpansion
set secsernum="0"

REM THE FOLLOWING SEARCHES FOR AN OS PARTITION AND GETS THE DEVICES SERIAL NUMBER
for /f %b in ('wmic volume get DriveLetter^, Label ^| find "UFD-Windows") do (
    for /f %l in ('powershell -File IronKey-GetSerialNumber.ps1 -driveLetter %b') do (
        if ERRORLEVEL 0 (
            set secsernum=%l
        )
    )
    echo Found OS partition on %b with serial number: !secsernum!
)
echo.
endlocal
REM *****
REM PART 2 END : GET SERIAL NUMBERS
REM *****

```

BEFORE YOU RUN THE SCRIPT

Make sure that the batch script folder (provision_batch_file_samples.zip) has been extracted from the main ironkeyutil1.5.zip package. The sample script provided is meant as a starting point from which you can create or customize a script that meets your specific provisioning requirements. At a minimum, you must modify the sample scripts to specify the following settings before running it:

Sample W300 script (ProvisionW300.cmd)

- Specify the path and filename of the Windows image file (WIM) that will be installed on the device; make sure that the WIM file is saved to the specified directory referenced in the script.
- Set the
- Set the BitLocker password for the device. Since W300 devices do not use hardware encryption, you can use Microsoft BitLocker software to encrypt the content on the OS partition.

Sample W500 script (ProvisionW500-W700.cmd)

- Specify the path and filename of the Windows image file (WIM) that will be installed on the device; make sure that the WIM file is saved to the specified directory referenced in the script.
- If you moved the IronKey Control Panel Installer folder from the default location in the Iron-Key_batch_sample_scripts folder, specify the new location. The script must be able to access the executable file to install the Control Panel on the device.
- Set the Admin Code and Management status parameters that will be applied to W500/W700 devices during initialization with IronKeyUtil.

To customize the W300 sample script

1. Open the provisionW300.cmd script file in a text editor with Administrator privileges.
2. Set the path to your WIM image by replacing the default path with the actual path and filename where your WIM file is located in the following section.

Default text: set mywim="C:\WIM\install.wim" Do not delete quotation marks.

```

REM This sets the path and file name for the WIM
set mywim="C:\WIM\install.wim"

```

3. Set the BitLocker password by replacing the default password "Welcome123" with the password you want to use. Do not delete quotation marks.

```
REM This sets the BitLocker password
REM NOTE: The standard BitLocker password requires the following:
REM      * at least 1 upper case character
REM      * at least 1 number
REM      * a minimum 8 characters
set bitLockerPassword="Welcome123"
```

4. Save the script and close the editor.

To customize the W500/W700 sample script

1. Open the provisionW500-W700.cmd script in a text editor with Administrator privileges.
2. Set the path to your WIM image by replacing the default path with the actual path and filename where your WIM file is located in the following section.

Default text: `set mywim="C:\WIM\install.wim"` Do not delete quotation marks.

```
REM This sets the path and file name for the WIM
set mywim="C:\WIM\install.wim"
```

3. If you moved the IronKey Control Panel Installer folder, specify the new folder location.

```
REM This sets the folder to the IronKey Control Panel
set cpInstallerFolder="IronKey Control Panel Installer"
```

4. In the section "PART 1 START : INITIALIZING DEVICES", search for and replace the following **init** command parameters that are required to initialize devices:
 - Default text: `set_admin_code=123123` Replace "123123" with the Admin Code you want to use.
 - Default text: `set_managed=0` The default setting is "0", or unmanaged. Type "1" to specify that the device will be managed by EMS.

```
REM This calls the IronKey Workspace Command Line Utility to:
REM      1) set the device admin code to "123123"
REM      2) initialize the device as an unmanaged device (0)
REM TODO: Modify set_admin_code and set_managed values
ironkeyutil.exe init drive_letter=%i set_admin_code=123123 set_managed=0
```

5. Save the script and close the editor.

RUNNING THE SAMPLE BATCH SCRIPT

Important: Make sure that you extract the `IronKey_batch_sample_scripts.zip` folder and can access it from the provisioning computer before you perform this procedure. Review the "Before you run the script" section prior to running the script.

To run the batch script

1. Log on to the provisioning computer running Windows 8.1 Enterprise.
2. Insert the IronKey Workspace devices that you want to provision into the available USB 3.0 ports or USB hub of the provisioning computer. **Note:** The sample scripts do not support provisioning W300 with W500/W700 devices in the same cycle.
3. Run a Windows command prompt as Administrator and change directories to the **IronKey_batch_sample_scripts** folder in the IronKeyUtil package.

```
C:\..\ironkeyutil1.5\sample scripts\IronKey_batch_sample_scripts
```

- At the command line prompt, type one of the following commands to start the batch script for the device type you are provisioning.

For W300 devices:

```
C:\..\ironkeyutil1.5\sample_scripts\IronKey_batch_batch_sample_scripts>provisionW300.cmd
```

For W500-W700 devices:

```
C:\..\ironkeyutil1.5\sample_scripts\IronKey_batch_batch_sample_scripts>provisionW500-W700.cmd
```

- Press **ENTER**.

Important: Do not unplug devices during the provisioning process. The time required will depend on the number of devices you are provisioning and the processor speed and USB port type being used.

- When the script finishes, review the command line output and IronKeyUtil log file (ikutil_YYMMDD_<computer name>.txt) to ensure that devices were configured correctly or to troubleshoot any errors during provisioning.
- After the cycle has finished, it is recommended that you allow a five minute cool-down period before removing devices. The provisioning process can warm devices to temperatures that may make them uncomfortable to handle immediately.
- Unplug all provisioned devices. If you want to start a new provisioning cycle, insert new devices and repeat steps 4 through 6.

Sample script output

The following code shows a sample of the on-screen output that will display at the command prompt during a provisioning cycle. The script created a device name using WTG and the serial number of the device. The device being provisioned in the output below is "WTG-02205945".

The items in bold below are the IronKey specific operations required for W500/W700 devices: Initializing the device, installing the IronKey Control Panel to the device, verifying that the IronKey Control Panel is successfully installed, and setting the device mode to Deployment. The final line of the script indicates that the devices are provisioned and ready to deploy.

```
This is a sample script provided by Imation.

Microsoft Windows 8.1 Enterprise detected...
Devices will be provisioned with the following image: "C:\Users\ca693306\Desktop\PS\WIM\install.wim"

Disabling Auto Play temporarily

Detecting and initializing devices
Please wait...

Found public partition on J:
Configuring J: device, please wait...

Re-enabling saved Auto Play settings

Devices are ready for image deployment

Devices being provisioned with the following image: "C:\Users\ca693306\Desktop\PS\WIM\install.wim"
Image: C:\Users\ca693306\Desktop\PS\WIM\install.wim

Id      Name                PSJobTypeName    State           HasMoreData     Location
--      -
5       Job5                 BackgroundJob     Running         True             localhost
5       Job5                 BackgroundJob     Completed       True             localhost
Log for G: workspace_info-G.log

Deployment Image Servicing and Management tool
Version: 6.3.9600.17031

Applying image
The operation completed successfully.
Set BitLocker default policies for WindowsToGo
Modify SAN Policy
Boot files successfully created.
Disk is now ready to be removed.
Provisioning completed in: 00:10:11.9120647 (hh:mm:ss.000)

Provisioning script complete.

Successfully provisioned device

Detecting OS partition, please wait...
Found OS partition on G:
Installing IronKey Control Panel to G:...
The IronKey Control Panel is now installed

Found public partition on J:
The J: serial number is: 11111111
Looking for OS partition on device serial number 11111111
Found OS partition on G: with serial number: 11111111
OS partition drive G: contains required IronKey components.

Device mode change takes effect the next time the device is plugged in. You must remove and re-plug
the device before running other commands on this device.

SUCCESS: Devices are provisioned and ready
```

PROVISIONING DEVICES USING SCCM

Microsoft Windows System Center Configuration Manager (SCCM) is a management tool for PCs and servers, to update software, set configuration and security policies, and monitor system status. SCCM can also create Windows To Go devices.

With the release of the IronKey Workspace Command Line Utility, you can now provision IronKey Workspace W500/W700 devices using SCCM in addition to W300s. When you deploy a provisioning package (with script and supporting files and applications) through SCCM, administrators or users can create Windows To Go devices from their client workstations. The sample scripts included with IronKeyUtil will provision W300, W500, and W700 devices and can be deployed through SCCM to create Windows To Go drives.

You must create a deployment package in SCCM that includes the provisioning script, applications and file dependencies. Before you create the package, make sure that your provisioning script and WIM image have been tested and modified according to your provisioning requirements. Whether you are using batch or PowerShell provisioning scripts, you must also ensure that SCCM and client workstations are properly configured before you push down the script package to the client system.

Client workstation requirements include:

- Computer running Windows 8.1 Enterprise. If creating a Windows 10 device, you must use Windows 10 Enterprise.
- Computer is joined to an Active Directory Domain
- USB hubs if provisioning multiple devices
- IronKey Workspace devices—W300, W500, and W700. W500/W700 devices must be at version 4.3, see “Supported devices” on page 5.

TROUBLESHOOTING

My drives are not recognized by the host computer

Verify that automount is enabled on the host system. The sample scripts will not function properly if this setting is disabled because Windows will not mount new basic and dynamic volumes or assign them drive letters. For information about how to enable/disable automount see, <https://technet.microsoft.com/en-us/library/cc753703.aspx>

Not enough drive letters

The scripts use drive letters from D to Z. Drives letters A to C are typically used by the host computer.

The sample PowerShell script allows you to provision W500/W700 and W300 drives in the same cycle. Keep this in mind when determining how many drives you can create in one cycle. If you want to provision more than the default number of drives that the script will allow, you will have to modify the script to handle drive letters dynamically as they become available during provisioning.

Error: You do not have Administrator rights to run this script!

All sample scripts must be run with Administrator privileges. For the Batch sample scripts, right-click the script file and click **Run as Administrator**. For PowerShell, run the PowerShell command prompt as Administrator and then run the script. You can also start the script from a Windows command prompt using the following parameters:

```
c:\>Powershell.exe -executionpolicy bypass -file "C:\ProvisionWorkspaceDevices.ps1"
```

If you are running a PowerShell command prompt (non-Admin), you can start the process to run PowerShell as Admin and then run the script.

```
PS> Start-Process powershell -Verb runAs
```