HPIA Driver Pushes

Contents

[Introduction 1](#_Toc113462457)

[Internet only version 2](#_Toc113462458)

[Model specific versions 2](#_Toc113462459)

[Deployment 2](#_Toc113462460)

[Package creation 3](#_Toc113462461)

[Required modules 3](#_Toc113462462)

[make\_new\_model\_collections.ps1 3](#_Toc113462463)

[HPIA\_sync\_drivers.ps1 3](#_Toc113462464)

[HPIA\_make Packages.ps1 4](#_Toc113462465)

[PowerShell App Deployment Toolkit 5](#_Toc113462466)

[Package deployment 5](#_Toc113462467)

[Plan for deployment 5](#_Toc113462468)

[Deployment group collections 6](#_Toc113462469)

[HPIA\_make\_collections&deployments.ps1 6](#_Toc113462470)

[Collection Clean up 8](#_Toc113462471)

[Deployment monitoring 9](#_Toc113462472)

[HPIA\_deployment\_status\_summary.ps1 9](#_Toc113462473)

[Spreadsheet tracking 9](#_Toc113462474)

[Status codes 10](#_Toc113462475)

[Deployment clean up 10](#_Toc113462476)

[HPIA\_deployment\_status\_build\_compliance.ps1 11](#_Toc113462477)

# Introduction

The HP Image Assistant (HPIA) tool is used to update firmware and drivers on our HP machines. It does this by scanning the machine and checking in with HP over the internet to see if the drivers on the machine are the latest for that model. We use this tool both during image deployment and as a stand-alone tool to make sure our machines stay up to date with drivers. This document is primarily concerned with the use of the packages in a push deployment. For the imaging use, see **Adding a New HP Model to the Image**.

## Internet only version

You can run HPIA completely over the internet and we even have a Configuration Manager package to do this as \Software Library\Overview\Application Management\Packages\OS Deployment\Drivers\HP Drivers\HPIA Installed\**HPIA Internet only**. This package has the executables and configuration files necessary to run HPIA but does not contain any drivers. An advantage of this package is that you can run it on any HP model. A disadvantage of this package is that all the needed drivers must be downloaded over the internet at runtime. It is not currently advertised to any collections.

## Model specific versions

Most of our HPIA packages are model specific, meaning that they contain the drivers for one model. You could include the drivers for more than one model, but then the package starts getting very large quickly as not too many SoftPaqs are shared among models. The advantage of this method is that most of the needed drivers are included in the package downloaded into the ConfigMgr cache folder before the HPIA tool is run so that it does not need to download them, and overall runtime is faster. A disadvantage to this is that there are many models to support so there are many packages that need maintenance on a regular basis.

## Deployment

Although we have been using HPIA to install drivers during imaging for a while, the first time we used the tool to push drivers to existing machine in the field was from September 2020 through February 2021. Because this was the first push of drivers and firmware to many machines, the push proceeded slowly. While later deployments will probably still include a firmware update, this is probably less likely, and we can probably ramp up the deployment faster. It has yet to be determined the how frequently we need to push these drivers; the current opinion is two or three times per year. As we get more experience with the tool, we may find that we need to update the drivers on specific models more frequently or in response to a security threat.

# Package creation

Package creation is generally performed as part of new model intake; see document **Adding a New HP Model to the Image**, HPIA Drivers section for more details. Specifically using PowerShell scripts **make\_new\_model\_collections.ps1** to make ConfigMgr collections, **HPIA\_sync\_drivers.ps1** to download drivers from HP to a file share, and **HPIA\_make Packages.ps1** to make the ConfigMgr packages for deployment.

### Required modules

The PowerShell script to synchronize with HP require the PowerShell module hp.repo from the HP Client Management Script Library, which you can download from here: [HP Client Management Script Library | HP Client Management Solutions](https://ftp.ext.hp.com/pub/caps-softpaq/cmit/hp-cmsl.html). Online documentation is available from there too. The specific commands used are from the SoftPaq Repository section.

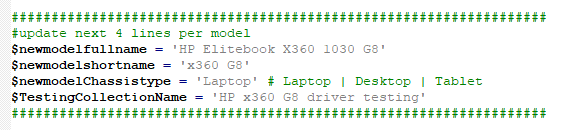
import-module hp.repo

The PowerShell scripts to work with ConfigMgr require the PowerShell module ConfigMgrPowerShellCmdlets, which is installed when the Configuration Manager console is installed. You can find more information about it here: [Configuration Manager PowerShell cmdlets - Configuration Manager | Microsoft Docs](https://docs.microsoft.com/en-us/powershell/sccm/overview?view=sccm-ps)

Import-Module "C:\Program Files (x86)\Microsoft Endpoint Manager\AdminConsole\bin\ConfigurationManager.psd1"

### make\_new\_model\_collections.ps1

In folder [\\server\share], open script file **make\_new\_model\_collections.ps1**. Edit these lines with model full name (from the CVA), model short name, chassis type, and testing collection name. SysID is used in the next script.



Then run it with an account that has permissions in the ConfigMgr console to create collections. It will create two collections for that model; one under All PCs | <chassis type> and one under Software Deployment\HP\driver testing that will be used in later scripts. Both collections will get membership rules.

### HPIA\_sync\_drivers.ps1

Then open script file **HPIA\_sync\_drivers.ps1**. Near the top is a section where the $HPModelsTable array is defined; you will need to add a new row to the table. Add a new row with the data from the CVA and collection script like this: (Yes, the OS is 2009 and not 20H2.) ProdCode is the SysID.

@{ ProdCode = '880D'; Model = "HP EliteBook 840 G8"; OSVER1 = 2009; OSVER2 = 1809; COLL = "HP 840 G8 driver testing" }

Comment out the other models you don’t need to synchronize with HP and run the script. It will create a new folder in [\\server\share] and populate it with files from HP, the App Deployment Toolkit **template** ( [\\server\share\ADTmaster] ) and the HPIA **template** ( [\\server\share\HPIA Base] ). You can see these template paths in the later part of the script using Robocopy. This script only copies the files from HP to a server share. This script also creates a drivers.csv file with a list of the drivers for that model that are part of the HPIA package, for reference purposes.

We will also need to run this script on a regular (quarterly?) basis for all models to keep the file share and packages up to date. The time and date stamp on the csv file can tell you when it was last synchronized.

### HPIA\_make Packages.ps1

Open the script file **HPIA\_make Packages.ps1**. Near the top is a section where the $HPModelsTable is defined; you will need to add a new row to the table. Add a new row with the data from the CVA and collection script like this: (Yes, the OS is 2009 and not 20H2.) Note that this table has an additional field (INC, for name of a collection to include) that the previous script did not have.

@{ ProdCode = '880D'; Model = "HP EliteBook 840 G8"; OSVER1 = 2009; OSVER2 = 1809; COLL = "HP 840 G8 driver testing"; INC = "All HP 840 G8 Laptops"}

Comment out the other models you don’t need to synchronize with the file share and run the script. It will create the package for the HPIA drivers in \Software Library\Overview\Application Management\Packages\OS Deployment\Drivers\HP Drivers\HPIA Installed from the source folder created and synchronized above. It will create 3 programs for the package (Driver Push, Imaging, and Self Install) and two advertisements of the Self Install program to the collections (COLL & INC) created above. It will replicate the package to the site distribution points for testing (make sure to replicate to All DPs before releasing to production). The package version is dated based on the activity log from the synchronization with HP, so it accurately reflects the files in the package.

Two of the programs (Imaging & Self Install) use the HP Image Assistant executable directly with this command line: Files\HPImageAssistant.exe /Operation:Analyze /Action:Install /Selection:All /Category:All /Silent /ReportFolder:"C:\Program Files (x86)\HP" /SoftpaqDownloadFolder:"Repository" /BIOSPwdFile:password /debug using either /Silent or /noninteractive as a user interface parameter. For more information about the parameters for HPImageAssistant.exe, see here: [HP Image Assistant User Guide](https://ftp.ext.hp.com/pub/caps-softpaq/cmit/whitepapers/HPIAUserGuide.pdf).

If the HPIA driver package already exists, it will be updated with the latest source files from the file share, the version updated, and synchronization to all the assigned distribution points will be triggered. Note that when replication is underway to a distribution point, imaging at that location is not available, so do not trigger the replication for existing packages used in the production image except during maintenance windows or when imaging is not in use. We do need to run this update replication on a regular (quarterly?) basis to keep all the packages used in the image up to date. Make sure you run this after the HP sync script above.

### PowerShell App Deployment Toolkit

The third program (Driver Push) uses both the HP Image Assistant and the PowerShell Application Deployment Toolkit. For more information on the settings for Deploy-Application.ps1, see here: [\\server\share\PSAppDeployToolkit.pdf]. The main reason to use the ADT is to get a welcome screen for the end user with a 6-hour countdown and a deferral button. The actual installation command is the same as for Self Install.

Graphical user interface, text

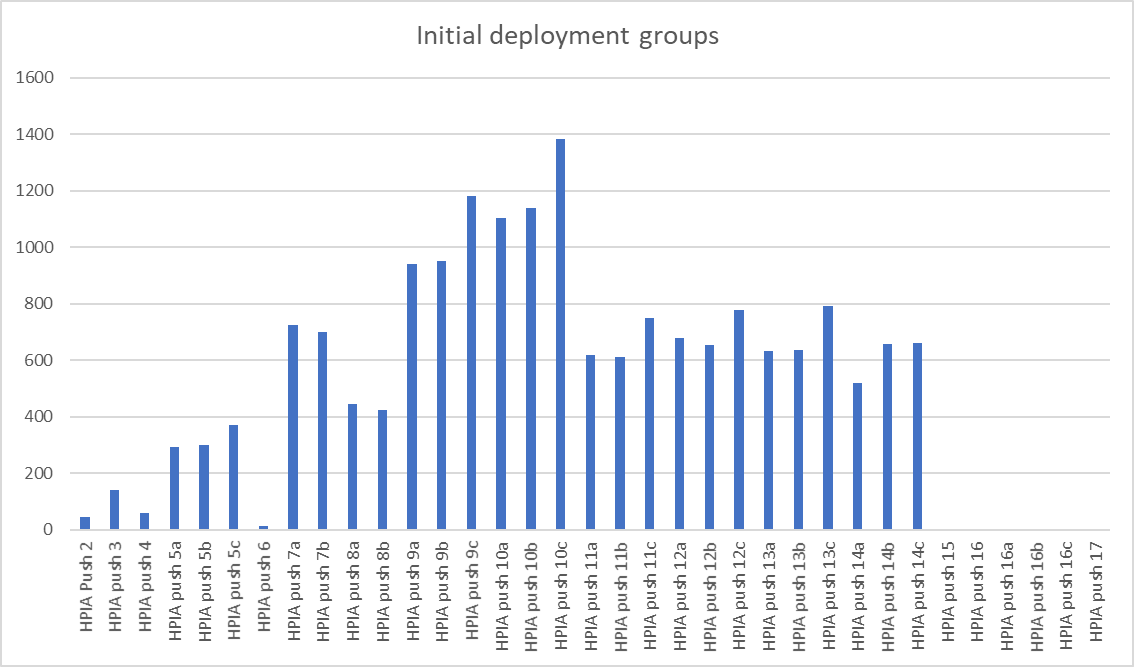
Description automatically generated

There is a sample package for this toolkit configuration here: [\\server\share\ADT\_deferrals] that does not install any actual drivers (runs command start-sleep -seconds 60) but does display all the ADT dialog boxes. That source folder is used for the ConfigMgr package Sample HPIA drivers so that pushing the drivers can also be simulated.

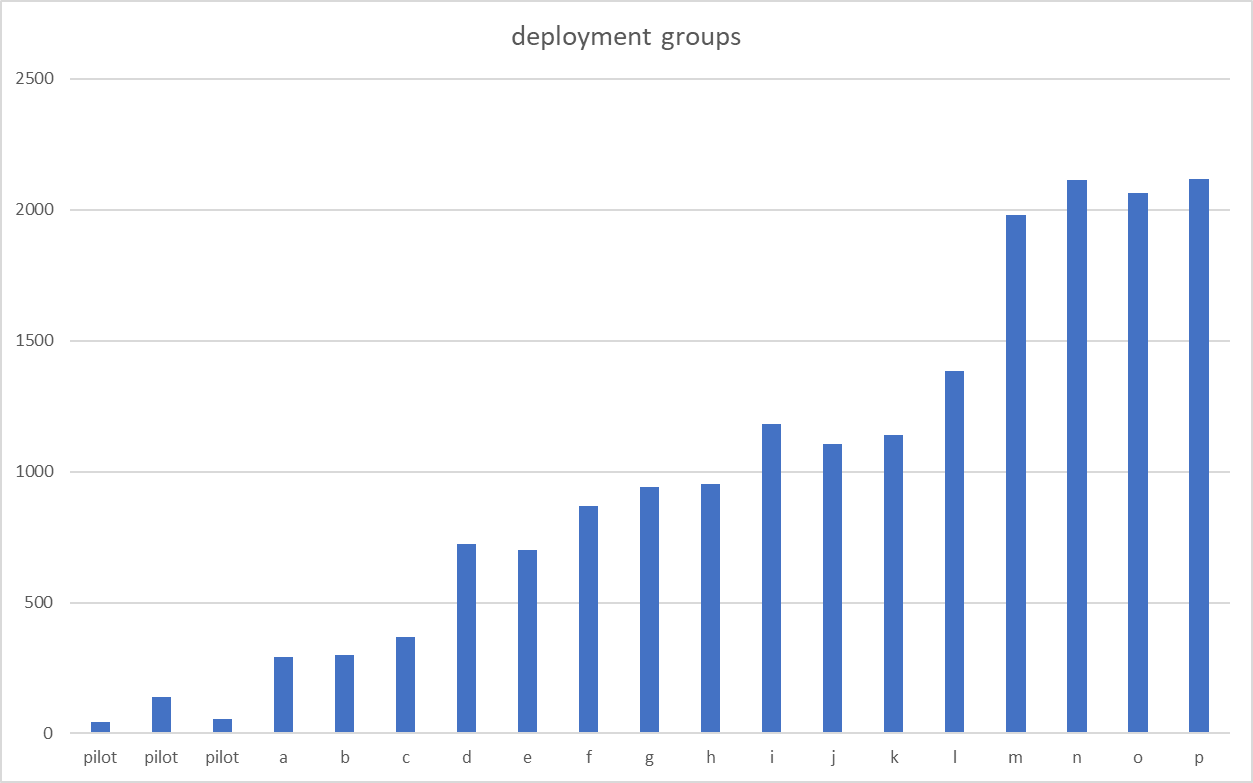
# Package deployment

## Plan for deployment

Plan for the deployment by selecting machine for pilot pushes to make sure the process is working as defined. In the past we have used these collections \Assets and Compliance\Overview\Device Collections\Software Deployment\HP\driver testing\**HPIA push 2** (team A), **HPIA push 3** (team B), & **HPIA push 4** (team C). Following organizational changes, we should probably revisit the definitions for those pilot collections (they are typically based on machines where the primary user is in one of the specified groups which in turn is based on direct membership and not any sort of dynamic org chart). After the pilot deployment, start the production deployment, beginning with smaller groups and ramping up to bigger and bigger collections. Within Device Collections\Software Deployment\HP\driver testing there are several collections that can be used named HPIA push 5 through HPIA push 14. Push 15 and 17 were used for compliance deployments and currently have no members; push 16 was used for a specific model that has since been merged into the main collections and currently has no members. In general, collections 7-10 are in the USA and randomized on the SMS GUID and collections 11-14 are at international locations and randomized on the SMS GUID. Within each are three sub-groups randomized on the last digit of the SMS GUID. But this makes for many small groups and will spread the deployment out over a long time.



We probably don’t need that many small groups and you can divide the population up any way you want. Here is an example of the deployment groups for August 2021, but even this is probably too many groups. It could probably be done with fewer, larger groups.



## Deployment group collections

Once you have decided on deployment groups, create ConfigMgr collections to reflect those groups. It is recommended you make the collections in \Assets and Compliance\Overview\Device Collections\Software Deployment\HP\driver testing and use a consistent naming standard.

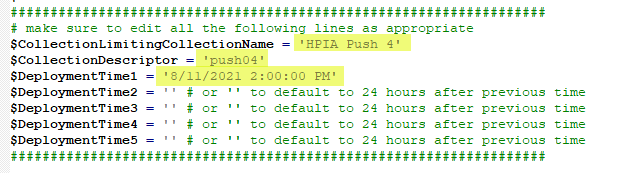
### HPIA\_make\_collections&deployments.ps1

Open the script file **HPIA\_make\_collections&deployments.ps1**. Near the top is a section where the $HPModelsTable is defined; you may need to add new rows to the table if new models have been added since the last time the script was run. Add a new row with the data from the CVA and collection script like this: (Yes, the OS is 2009 and not 20H2.) Note that this table also has the additional field (INC, for collection to include).

@{ ProdCode = '880D'; Model = "HP EliteBook 840 G8"; OSVER1 = 2009; OSVER2 = 1809; COLL = "HP 840 G8 driver testing"; INC = "All HP 840 G8 Laptops"}

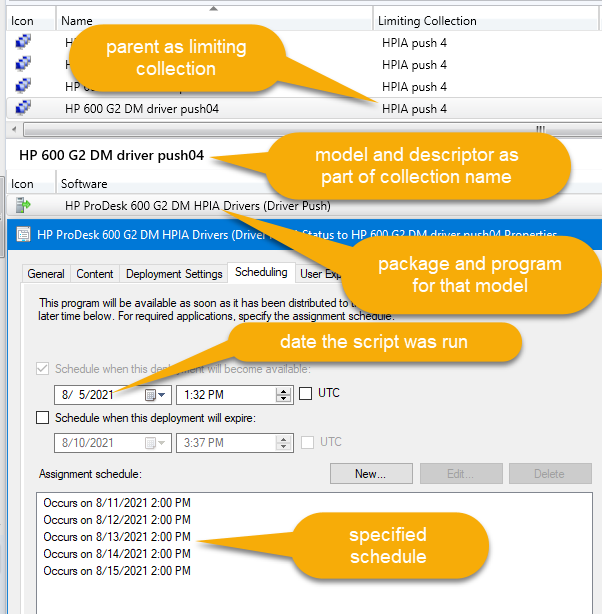
This time do not comment out any of the model rows as we will need them all, unless you know for certain that your deployment collection doesn’t have any of that model. It’s easier to leave them in and simply delete the resulting collections later.

Instead fill in these lines with information about the specific deployment group: Parent collection name, advertisement descriptor, and push start time and date.



The script will make many collections in \Assets and Compliance\Overview\Device Collections\Software Deployment\HP\driver pushes for this deployment, one collection for each model in the table using the parent collection as the limiting collection and the descriptor as part of the name after the model. It will create an advertisement of the appropriate model package to that collection with a set of five mandatory deployment times (because users can defer the deployment and/or their machines may be off at the scheduled time). You must specify the first deployment time; it is optional to specify the other times. If you don’t specify the later deployment times, it will add 24 hours to the previous time and use that, which means it will include weekends.

Because the collections created use an include rule (the INC field from the table), it does not have an evaluation schedule.

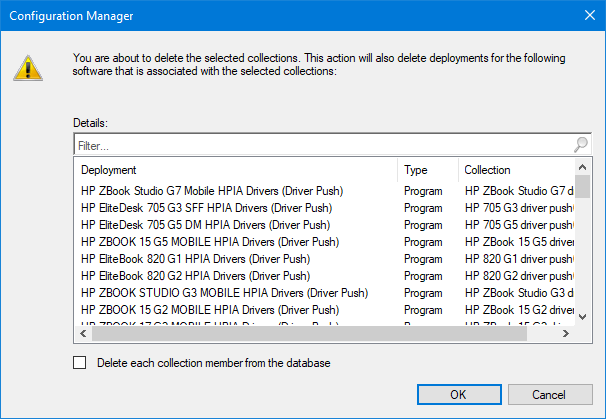


### Collection Clean up

After the script has run and created all the deployment collections, and after Configuration manager has had a chance to populate members into them (~2-24 hours), you can delete the deployment collections that have no members to free up ConfigMgr resources.

Graphical user interface, application

Description automatically generated



# Deployment monitoring

After the collections and deployments are created, it’s probably a good idea to monitor their status.

### HPIA\_deployment\_status\_summary.ps1

Open the script file **HPIA\_deployment\_status\_summary.ps1**. Near the top is a section where the $HPModelsTable is defined; you may need to add new rows to the table if new models have been added since the last time the script was run. Add a new row with the data from the CVA and collection script like this: (Yes, the OS is 2009 and not 20H2.) Note that this table also has the additional field (INC, for collection to include).

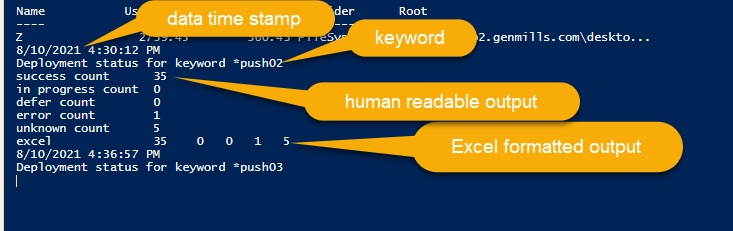
@{ ProdCode = '880D'; Model = "HP EliteBook 840 G8"; OSVER1 = 2009; OSVER2 = 1809; COLL = "HP 840 G8 driver testing"; INC = "All HP 840 G8 Laptops"}

This time do not comment out any of the model rows as we will need them all.

Instead, look for the line defining the variable $keywords. Populate the variable with the appropriate deployment descriptor form when the deployment collections were created. Since $keywords can be an array, you can include multiple deployments in the same script.

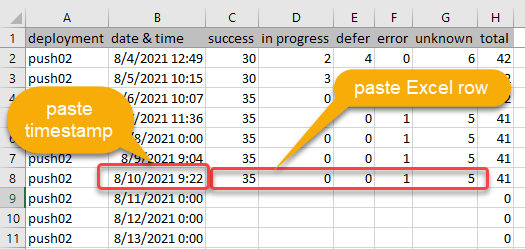
$keywords = "\*push02","\*push03","\*push04"

When you run the script, it will look for the advertisements using the keyword and the model name and gather summary data for the status. It will present total counts for successful deployments, deployments in progress, number deferred, number of errors and number of unknown systems. It will present the numbers both as a list and in a single row for easy pasting into an Excel spreadsheet.



### Spreadsheet tracking

It is recommended you use a spreadsheet (or database) to track the performance of the deployments over time because ConfigMgr status is not reported over time and there are a large number of deployments (one per model) for each push. The output of the script is formatted for easy use in an Excel spreadsheet.



### Status codes

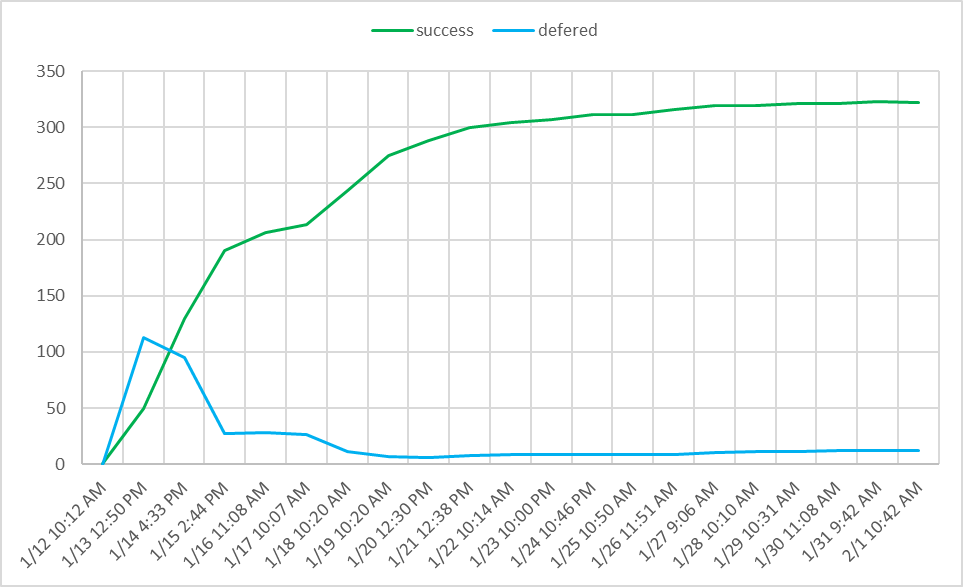
The status message ID is used to determine the status for each deployment. Near the beginning of the script are the known codes for success ($successcodes), in progress ($inprogresscodes), error ($errorcodes), & unknown ($unknowncodes). If a new status code is returned, the script will report this in the output, and you will need to determine what it means and add it to the script.

{write-host "\*\*\* new status found $status.CollectionName, $status.MessageID, $status.MessageDescription"}

Since we are using the App Deployment Toolkit with the ability to defer there is special handling for users that deferred. Normally the status message ID of 10006 would mean failure, so we then check the return code for 60012 or 256, which meant the user deferred the deployment. Because an ADT deferral is reported to ConfigMgr as a failure, that is why the advertisement has multiple mandatory deployment times, so that it can be attempted again.

# Deployment clean up

In the past, our deployments have had a success rate of above 80%, and generally reach 75%-80% in about 10 days. We typically leave the advertisement in place for 3 weeks; after that few machines run it. It’s better to collect the names of the machines that have not run it or failed and add them to a compliance collection for a second push of the driver package.



Once you have collected the names (script below), you can stop monitoring the deployment and delete the advertisements and collections to save ConfigMgr resources.

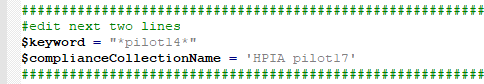
### HPIA\_deployment\_status\_build\_compliance.ps1

Open the script file **HPIA\_deployment\_status\_build\_compliance.ps1**. Near the top is a section where the $HPModelsTable is defined; you may need to add new rows to the table if new models have been added since the last time the script was run. Add a new row with the data from the CVA and collection script like this: (Yes, the OS is 2009 and not 20H2.) Note that this table also has the additional field (INC, for collection to include).

@{ ProdCode = '880D'; Model = "HP EliteBook 840 G8"; OSVER1 = 2009; OSVER2 = 1809; COLL = "HP 840 G8 driver testing"; INC = "All HP 840 G8 Laptops"}

This time do not comment out any of the model rows as we will need them all, unless you know for certain that your deployment collection doesn’t have any of that model. It’s easier to leave them in and simply delete the resulting collection later.

Instead fill in these lines with information about the specific deployment: keyword and new compliance collection name.



The script will review the results of the deployments with the keyword (just like HPIA\_deployment\_status\_summary.ps1) but for those machines that are listed as in progress or error, it will add the machine to a new collection for a new compliance push. This new collection will then be the limiting collection for a run of HPIA\_make\_collections&deployments.ps1.