Marable, Mike

Explanation of how the WaaS process is structured within Configuration Manager.

WaaS CM Structure

Version 4

Contents

[WaaS: Script and Configuration Files 1](#_Toc11754632)

[Core Buildout Script 1](#_Toc11754633)

[Configuration XML Files 1](#_Toc11754634)

[Console Folder Structure 4](#_Toc11754635)

[WaaS: Pre-Assessment Phase 4](#_Toc11754636)

[Core Collections 4](#_Toc11754637)

[Ready for Pre-Assessment 4](#_Toc11754638)

[Passed Pre-Assessment 4](#_Toc11754639)

[Reference Collections 5](#_Toc11754640)

[Exclusion Collections 5](#_Toc11754641)

[Hardware Collections 6](#_Toc11754642)

[Operating System Collections 6](#_Toc11754643)

[Remediation Collections 7](#_Toc11754644)

[BitLocker 7](#_Toc11754645)

[Disk Space 8](#_Toc11754646)

[Outdated Hardware Inventory 8](#_Toc11754647)

[Unsupported Hardware 9](#_Toc11754648)

[WaaS: CompatScan Phase 9](#_Toc11754649)

[WaaS: In-Place Upgrade Phase 9](#_Toc11754650)

# WaaS: Script and Configuration Files

There is a build-out script and some XML configuration files that automates the creation of the collections and deployments used in the WaaS process.

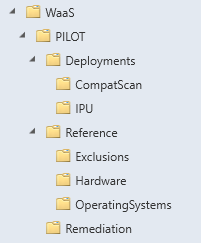
## Core Buildout Script

|  |  |
| --- | --- |
| Script Name: | WaaS\_))\_Staging-Core\_v#.#.ps1 |

This script requires 2 command line parameters.

|  |  |  |
| --- | --- | --- |
| Parameter | Possible Values | Purpose |
| WaaSBranch | DEV/QA/PILOT/PROD | Identifies what release stage the WaaS buildout should use. [See screenshot below] |
| BuildNum | 1709/1803/1809… | What build number should be used for the buildout. |

The WaaSBranch value will be used to create the parent folder within the WaaS root folder in the console. It is also used in the naming convention of all of the collections.

Example: WaaS\_**PILOT**\_1809\_01\_Ready\_for\_PreAssessment

The BuildNum value will also be used in the naming convention of all of the collections. It will also be used in the OS reference collections to identify which Windows 10 builds are prior and eligible to be upgraded and which are current or newer. It finally will be used to identify which CompatScan sequence should be deployed to the CompatScan collection.

Example: WaaS\_PILOT\_**1809**\_01\_Ready\_for\_PreAssessment

This allows for running multiple WaaS tracks in parallel. For example, you could have one WaaS track for 1809 in PROD and at the same time a second 1809 track in DEV. Or an 1809 in PROD and a 1903 in DEV.

## 

## Configuration XML Files

There are 4 configuration files that the script leverages.

1. Config.xml

This file defines the sitecode and site server name

1. BuildNumbers.xml

This file lists the XREF between the “friendly” build number (i.e. 1709/1809/etc.) with the corresponding version number (i.e. 16299/17763/etc.)

1. <buildNum>.xml

There will be an XML file named after each ‘friendly” build number (i.e. 1809.xml). The file will provide the Task Sequence IDs for the CompatScan and IPU sequences for each of the 4 branches (DEV/QA/PILOT/PROD) as well as the prior builds.

1. Windows10Hardware.csv

This CSV contains the information on the supported hardware models

|  |  |
| --- | --- |
| Sample Files | |
| Config.xml | <Settings>      <SiteInfo>          <SiteCode>MM1</SiteCode>          <ServerName>UHCMPRISPR1</ServerName>      </SiteInfo>  </Settings> |
| BuildNumbers.xml | <BuildNumbers>  <Build ID="1507">  <Ver>10240</Ver>  </Build>  <Build ID="1511">  <Ver>10586</Ver>  </Build>  <Build ID="1607">  <Ver>14393</Ver>  </Build>  <Build ID="1703">  <Ver>15063</Ver>  </Build>  <Build ID="1709">  <Ver>16299</Ver>  </Build>  <Build ID="1803">  <Ver>17134</Ver>  </Build>  <Build ID="1809">  <Ver>17763</Ver>  </Build>  <!-- Build 19H1 -->  <Build ID="1900">  <Ver>18272</Ver>  </Build>  <Build ID="1903">  <Ver>18362</Ver>  </Build>  </BuildNumbers> |
| 1809.xml | <Settings>  <TSIDs>       <DEV>  <CompatScan>MM1005C6</CompatScan>  <IPU>MM1005C7</IPU>       </DEV>       <QA>  <CompatScan>MM1004E2</CompatScan>  <IPU>MM1004E3</IPU>       </QA>       <PILOT>  <CompatScan>MM100641</CompatScan>  <IPU>MM100646</IPU>       </PILOT>       <PROD>  <CompatScan>MM1004E2</CompatScan>  <IPU>MM1004E3</IPU>       </PROD>  </TSIDs>  <PreviousBuilds>1507,1511,1607,1703,1709,1803</PreviousBuilds>  </Settings> |
| Windows10Hardware.csv | Baseboard,Model,List,Branch  ,Surface Book,W,PROD  ,Surface Pro 4,W,PROD  ,Virtual Machine,W,PROD  ,VMware Virtual Platform,W,PROD  ,VMware71,W,PROD  03441V,Latitude 5590,G,DEV  18E5,HP EliteDesk 800 G1 USDT,W,PROD  18E7,HP ProDesk 600 G1 SFF,W,PROD |

# WaaS: Pre-Assessment Phase

## Core Collections



### Ready for Pre-Assessment

|  |  |
| --- | --- |
| Limiting Collection | Eligible Computers |
| Update Time | Daily @ 1:00 am |

This is the entry point for the WaaS process. Membership in this collection can be through any method, direct membership, inclusion collections or queries. The sole purpose of this collection is to act as the “starting line” for the entire process.

### Passed Pre-Assessment

|  |  |
| --- | --- |
| Limiting Collection | Eligible Computers |
| Update Time | Daily @ 2:00 am |

As the name implies, machines that pass the pre-assessment tests are collected here and are ready to move on to the next phase (CompatScan).

Membership is controlled with a series of Include and Exclude Collection rules. The Exclude Collection rules will be detailed later.

|  |  |
| --- | --- |
| Collection | Include / Exclude |
| Ready for Pre-Assessment | Include |
| BitLocker Remediation | Exclude |
| Disk Space Remediation | Exclude |
| Outdate Hardware Inventory | Exclude |
| Unsupported Hardware | Exclude |
| “All Exclusions” | Exclude |

The logic works like this, all of the machines flow into this “passed” collection. Machines found to have failed any of the pre-assessment tests will fall into one or more of the remediation collections and will be filtered out. Finally, any machines which are explicitly excluded fill funnel into the “All Exclusions” collection and be filtered out.

More details on these exclude collections follows.

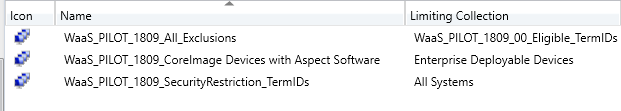
## Reference Collections

Under the “branch” folder is a subfolder named “Reference”. Inside this folder will be collections used to gather machines into various categories. For example, collections breaking down the various operating systems found in the environment.

### Exclusion Collections

This folder will contain collections used to explicitly exclude machines for any number of reasons. If there is a particular application that will prevent the upgrade, a collection used to gather machines with that application will be created here. Or, you could create a collection of machines belonging to the executives, or critical devices that require extra care when upgrading.

All of these collections funnel up into a master “All Exclusions” collection. In turn, this master collection is used to exclude devices from the “Passed Pre-Assessment” collection as noted previously.



The individual exclusion collections can be constructed in whatever manner meets the need. They all will be added as an Include Collection rule to the “All Exclusions” collection.

#### All Exclusions Collection

|  |  |
| --- | --- |
| Limiting Collection | Eligible Computers |
| Update Time | Daily @ 6:00 am |

The purpose of this collection is to act as a master repository of all devices that are excluded for whatever reason to be prevented from moving beyond the Pre-Assessment phase of the process.

### Hardware Collections

The collections in this folder are meant to filter out machines using unsupported hardware. Not all Windows 7 devices are going to be supported by the organization for Windows 10. Eventually there will be models that are retired and not going to be supported on newer Windows 10 builds.

There is a collection for each hardware model supported. Being a primarily an HP shop, the queries identify models based on their baseboard identifier. In the case of non-HP models (i.e. the Microsoft Surface line) the model name is used.

#### Example: HP Elitebook 840 G1

|  |  |
| --- | --- |
| Limiting Collection | All Systems |
| Update Time | Daily @ 1:15 am |

|  |
| --- |
| Query: |
| *select SMS\_R\_SYSTEM.ResourceID,SMS\_R\_SYSTEM.ResourceType,SMS\_R\_SYSTEM.Name,SMS\_R\_SYSTEM.SMSUniqueIdentifier,SMS\_R\_SYSTEM.ResourceDomainORWorkgroup,SMS\_R\_SYSTEM.Client from SMS\_R\_System inner join SMS\_G\_System\_BASEBOARD on SMS\_G\_System\_BASEBOARD.ResourceId = SMS\_R\_System.ResourceId where SMS\_G\_System\_BASEBOARD.Product = "198F"* |

All of the individual model collections funnel up into a master “All Supported Hardware” collection. This then is used to identify devices running on unsupported hardware, which will be discussed in the “Remediation Collections” section.

### Operating System Collections

The collections in this folder are intended to gather up the various Operating Systems in the environment.

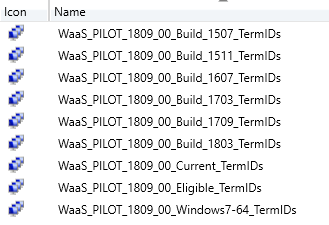
|  |  |
| --- | --- |
| Limiting Collection | All Systems |
| Update Time | Daily @ 12:00 am |

Example: Windows 10 – 1809

|  |
| --- |
| Query: |
| *select SMS\_R\_SYSTEM.ResourceID,SMS\_R\_SYSTEM.ResourceType,SMS\_R\_SYSTEM.Name,SMS\_R\_SYSTEM.SMSUniqueIdentifier,SMS\_R\_SYSTEM.ResourceDomainORWorkgroup,SMS\_R\_SYSTEM.Client from SMS\_R\_System inner join SMS\_G\_System\_OPERATING\_SYSTEM on SMS\_G\_System\_OPERATING\_SYSTEM.ResourceId = SMS\_R\_System.ResourceId where SMS\_G\_System\_OPERATING\_SYSTEM.BuildNumber = '17134'* |

Each of these OS collections funnels up into 1 of 2 master OS collections:

1. Eligible Computers
2. Current Computers



Prior versions of Windows that are eligible to upgrade will fall into the “Eligible” collection. This is the limiting collection for the various “Core” collections (i.e. the Ready for Pre-Assessment and Passed Pre-Assessment). This ensures that only devices running a prior version of Windows (one that is eligible to upgrade) fall into the collections.

## Remediation Collections

These collections are intended to gather up devices that do not pass the various pre-assessment tests.

### BitLocker

Identifying macines that are not encrypted with BitLocker is a little tricky. If a device does not have BitLocker enabled, the WMI class does not exist and you cannot query for a non-existent class. So we first identify compliant systems and use that to find non-compliant systems.

#### Compliant BitLocker

|  |  |
| --- | --- |
| Limiting Collection | Ready for Pre-Assessment |
| Update Time | Daily @ 1:15 am |

First we query for all devices that are properly BitLocker encrypted.

|  |
| --- |
| Query: |
| *select SMS\_R\_SYSTEM.ResourceID,SMS\_R\_SYSTEM.ResourceType,SMS\_R\_SYSTEM.Name,SMS\_R\_SYSTEM.SMSUniqueIdentifier,SMS\_R\_SYSTEM.ResourceDomainORWorkgroup,SMS\_R\_SYSTEM.Client from SMS\_R\_System inner join SMS\_G\_System\_BITLOCKER\_DETAILS on SMS\_G\_System\_BITLOCKER\_DETAILS.ResourceId = SMS\_R\_System.ResourceId where SMS\_G\_System\_BITLOCKER\_DETAILS.Compliant = 1* |

This collection is then used as an Exclusion Collection in the remediation collection.

#### Remediate BitLocker

|  |  |
| --- | --- |
| Limiting Collection | Ready for Pre-Assessment |
| Update Time | Daily @ 1:30 am |

This collection uses a pair of collection rules for its membership.

|  |  |
| --- | --- |
| Collection | Include / Exclude |
| Ready for Pre-Assessment | Include |
| Compliant BitLocker | Exclude |

The result is a collection of machines that have, for any reason, failed to report that they are properly BitLocker encrypted.

This is then uses as an Exclusion Collection rule on the “Passed Pre-Assessment” collection.

### Disk Space

|  |  |
| --- | --- |
| Limiting Collection | Ready for Pre-Assessment |
| Update Time | Daily @ 1:35 am |

This collection will identify devices that have less than 25GB of free disk space on the C: drive.

|  |
| --- |
| Query: |
| *select SMS\_R\_SYSTEM.ResourceID,SMS\_R\_SYSTEM.ResourceType,SMS\_R\_SYSTEM.Name,SMS\_R\_SYSTEM.SMSUniqueIdentifier,SMS\_R\_SYSTEM.ResourceDomainORWorkgroup,SMS\_R\_SYSTEM.Client from SMS\_R\_System inner join SMS\_G\_System\_LOGICAL\_DISK on SMS\_G\_System\_LOGICAL\_DISK.ResourceId = SMS\_R\_System.ResourceId where SMS\_G\_System\_LOGICAL\_DISK.DeviceID = 'C:' and SMS\_G\_System\_LOGICAL\_DISK.FreeSpace < 25* |

This collection is then set as an Exclusion Collection to the “Passed Pre-Assessment” collection. This then removes any devices with less than 25GB free from the passed collection.

### Outdated Hardware Inventory

|  |  |
| --- | --- |
| Limiting Collection | Ready for Pre-Assessment |
| Update Time | Daily @ 1:50 am |

This collection will identify devices that have not reported hardware inventory within the last 14 calendar days.

|  |
| --- |
| Query: |
| *select SMS\_R\_SYSTEM.ResourceID,SMS\_R\_SYSTEM.ResourceType,SMS\_R\_SYSTEM.Name,SMS\_R\_SYSTEM.SMSUniqueIdentifier,SMS\_R\_SYSTEM.ResourceDomainORWorkgroup,SMS\_R\_SYSTEM.Client from SMS\_R\_System inner join SMS\_G\_System\_WORKSTATION\_STATUS on SMS\_G\_System\_WORKSTATION\_STATUS.ResourceId = SMS\_R\_System.ResourceId where SMS\_G\_System\_WORKSTATION\_STATUS.LastHardwareScan < DateAdd(dd,-14,GetDate())* |

This collection is then set as an Exclusion Collection to the “Passed Pre-Assessment” collection to remove these outdated devices from the process.

### Unsupported Hardware

|  |  |
| --- | --- |
| Limiting Collection | Ready for Pre-Assessment |
| Update Time | Daily @ 1:35 am |

This collection identifies machines that are running on unsupported hardware models. It’s membership is controlled by a pair of collection rules.

|  |  |
| --- | --- |
| Collection | Include / Exclude |
| Ready for Pre-Assessment | Include |
| All Supported Hardware | Exclude |

The end result is a collection of devices that are running on unsupported models. This is then set as an Exclusion Collection on the “Passed Pre-Assessment” collection.

# WaaS: CompatScan Phase

# WaaS: In-Place Upgrade Phase