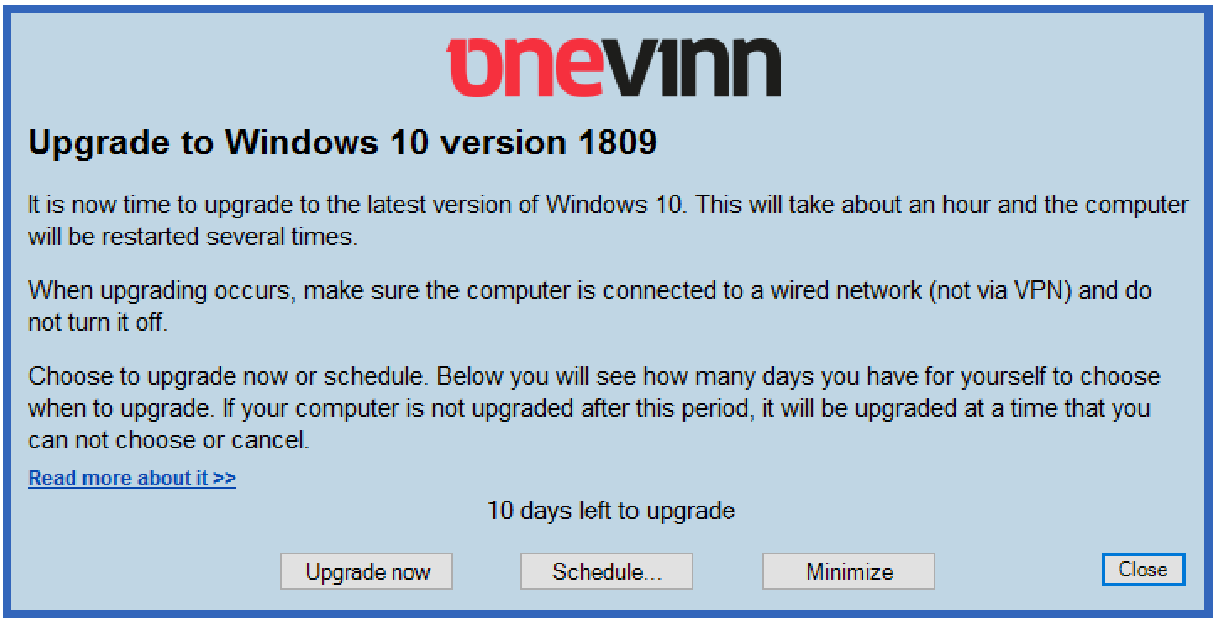
Onevinn Windows 10 Upgrade Tools



Upgrade to Windows 10 version 1809

It is now time to upgrade to the latest version of Windows 10. This will take about an hour and the computer will be restarted several times.

When upgrading occurs, make sure the computer is connected to a wired network (not via VPN) and do not turn it off.

Choose to upgrade now or schedule. Below you will see how many days you have for yourself to choose when to upgrade. If your computer is not upgraded after this period, it will be upgraded at a time that you can not choose or cancel.

Read more about it >>

10 days left to upgrade

CONTENTS

1. BACKGROUND

Most experts agree that the best way to upgrade Windows 10 in the enterprise is through a SCCM Task Sequence. No other means allows the same level of customizations.

However most suitable, a task sequence has its flaws and especially the end user experience could be improved. The **Onevinn Windows 10 Upgrade Tools** are intended to fill this gap by both allowing the end user to schedule his or her upgrade down to the minute and improving the esthetic experience by showing a custom background throughout the process.

The Custom background (UPGBackground.exe) not only looks nice(r) it also effectively prevents any premature logon attempt as well as adds password protected debug functionality to the Task Sequence.

In this, the second, release of "The tools" a console extension has been added to allow for central monitoring of the upgrades.

2. VERSION

Version	Author	Date	Remark
1.0	Johan Schrewelius	2018-04-15	Document created
1.1	Johan Schrewelius	2018-05-30	New minor release
2.0	Johan Schrewelius	2018-09-23	Version 2.0 released
2.1	Johan Schrewelius	2018-10-28	Hotfixes and added functionality.
2.2	Johan Schrewelius	2018-11-19	Check Changelog for details.
2.3	Johan Schrewelius	2020-06-29	Check Changelog for details.
2.4	Johan Schrewelius	2021-01-18	AC check and Toast added. bugfixes

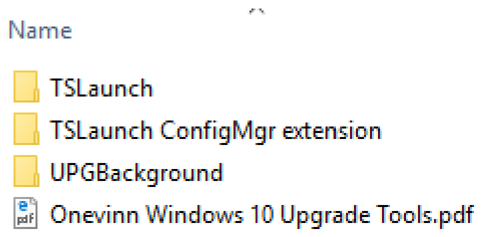
3. COMPLEXITY

TSLaunch, the upgrade UI / scheduler, has become extremely versatile with numerous different configuration alternatives. It's obvious that all alternatives and options cannot be combined. You will have to consider this when creating your own configuration.

Always make sure that you have an already tested and fully functional upgrade task sequence before adding TSLaunch to the equation, it could easily get out of hand otherwise.

4. COMPONENTS

Once downloaded and unpacked you will have three folders and this document:



Name	Type
TSLaunch	File folder
TSLaunch ConfigMgr extension	File folder
UPGBackground	File folder
Onevinn Windows 10 Upgrade Tools.pdf	PDF File

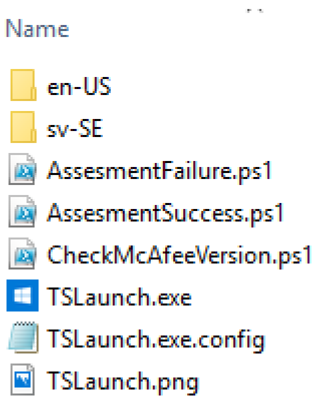
TSLaunch contains a Task Sequence launcher application, **TSLaunch.exe**, a configuration file, two language folders, an example logo and example scripts.

TSLaunch ConfigMgr extension contains a single msi to extend the ConfigMgr console with a new monitoring node.

UPGBackground contains a msi and an example language file. The msi installs, apart from UPGBackground.exe, a simple windows service that is responsible for launching the background application at the earliest possible moment after every reboot.

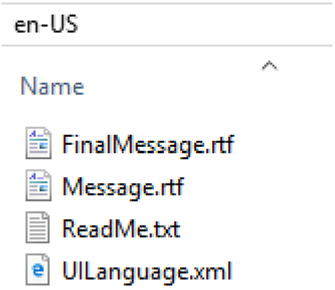
5. FILES

TSLaunch is designed to be deployed to the same computers as an upgrade TS. The TS in question should be deployed in such a way that it is never automatically launched by other means than TSLaunch. This can be achieved by deploying it as “Available” or by scheduling it (“Required”) far into the future, long after it has turned irrelevant. The choice depends on whether visibility in Software Center is desired. TSLaunch supports multiple languages, it falls back to en-US if it the detected System UI Language is not found in the folder structure. You can add as many language folders as required. For clarity check the included example files.



Name

- en-US
- sv-SE
- AssesmentFailure.ps1
- AssesmentSuccess.ps1
- CheckMcAfeeVersion.ps1
- TSLaunch.exe
- TSLaunch.exe.config
- TSLaunch.png



en-US

Name

- FinalMessage.rtf
- Message.rtf
- ReadMe.txt
- UILanguage.xml

File	Location	Usage
TSLaunch.exe	Root	The application it-self
TSLaunch.exe.config	Root	Configuration
TSLaunch.png	Root	Logo (png/jpg)
UILanguage.xml	en-US	UI text, buttons etc.
Message.rtf	en-US	Standard message to user.
FinalMessage.rtf	en-US	Stronger message, shown instead of standard when time is running short. Can contains same text as Message.rtf.
CheckMcAfeeVersion.ps1 AssesmentFailure.ps1 AssesmentSuccess.ps1	Root	Example scripts. Explained further down in this guide.

Three variables are available for use in the message files:

%USERNAME% - Resolves to user logon name
%COMPUTERNAME% - Resolves to computername
%DAYSLEFT% - Remaining days before the deployment reaches its end date.

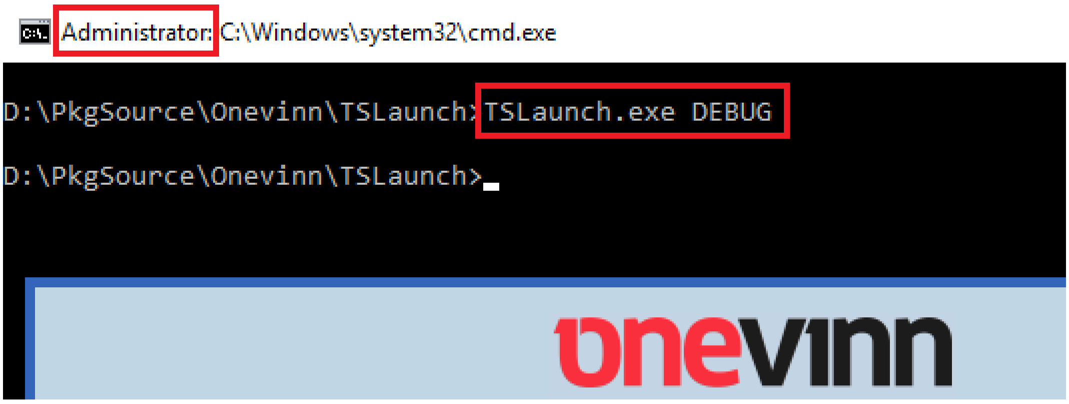
6. DESIGN MODE

To ease the design work TSLaunch kan be run in debug (design) mode.

Start the application from a command prompt (**Admin**):

TSLaunch.exe DEBUG (an otherwise hidden close button will be available in this mode).

Remark: The included en-US translation was made using a well known online service which's quality could be disputed.



```
Administrator: C:\Windows\system32\cmd.exe

D:\PkgSource\Onevinn\TSLaunch>TSLaunch.exe DEBUG

D:\PkgSource\Onevinn\TSLaunch>_
```

7. LOGGING

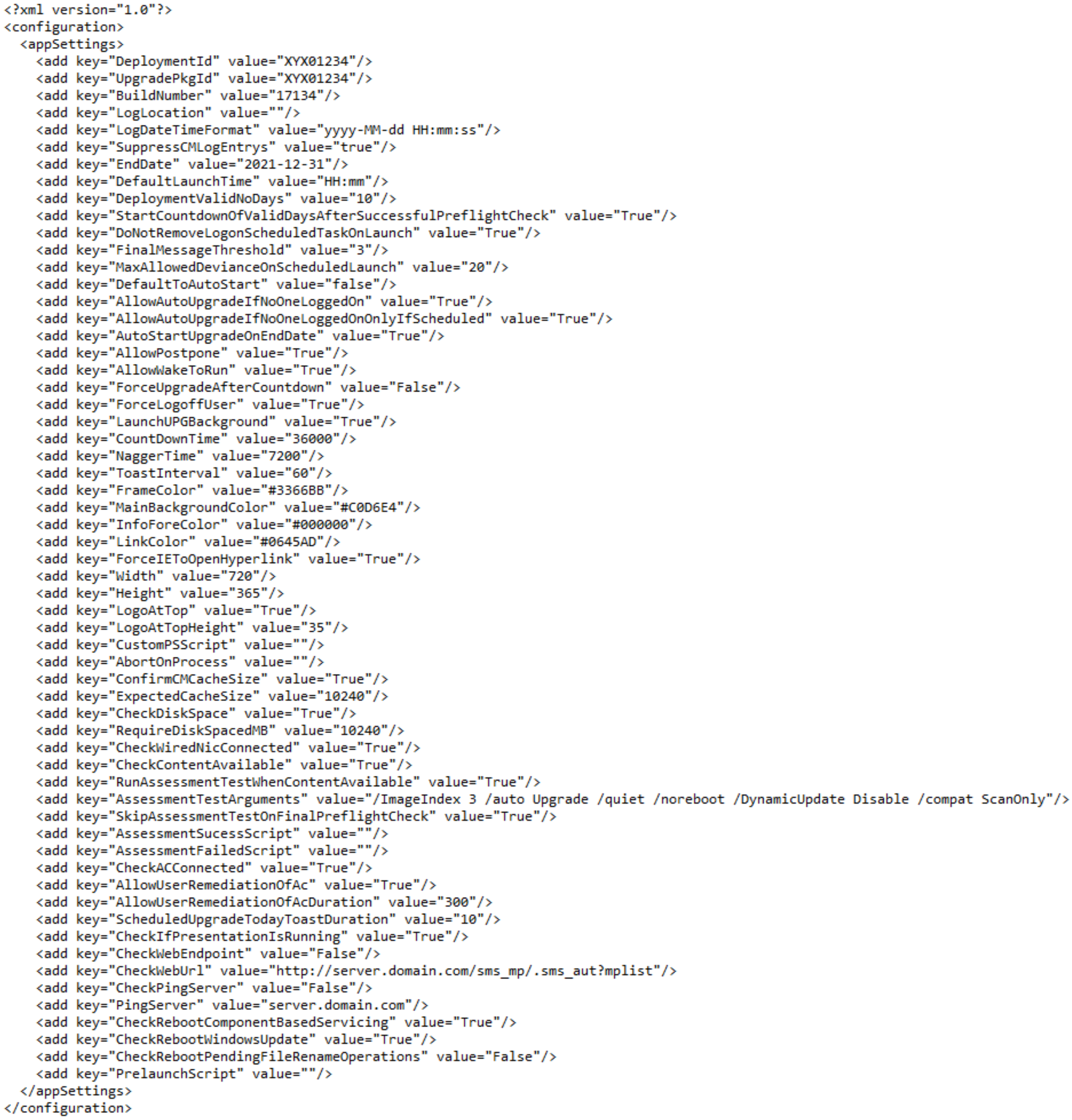
TSLaunch creates and uses a local log file at "%WinDir%\Temp\TSLaunch.log". This is the main source for information during setup and design.

In the TSLaunch.exe.config file it's possible to indicate a remote log location (file share). TSLaunch will maintain a copy of the local log at this location, the name annotation for these files are: <ComputerName>_<BuildNo>_TSLaunch.log

TSLaunch is also sending status messages to ConfigMgr, a feature that is covered in a later section of this manual.

8. TSLAUNCH.EXE.CONFIG

This is where all behavioral configuration is done. The file also holds information about which TS to launch and pre-requisites for allowing it.



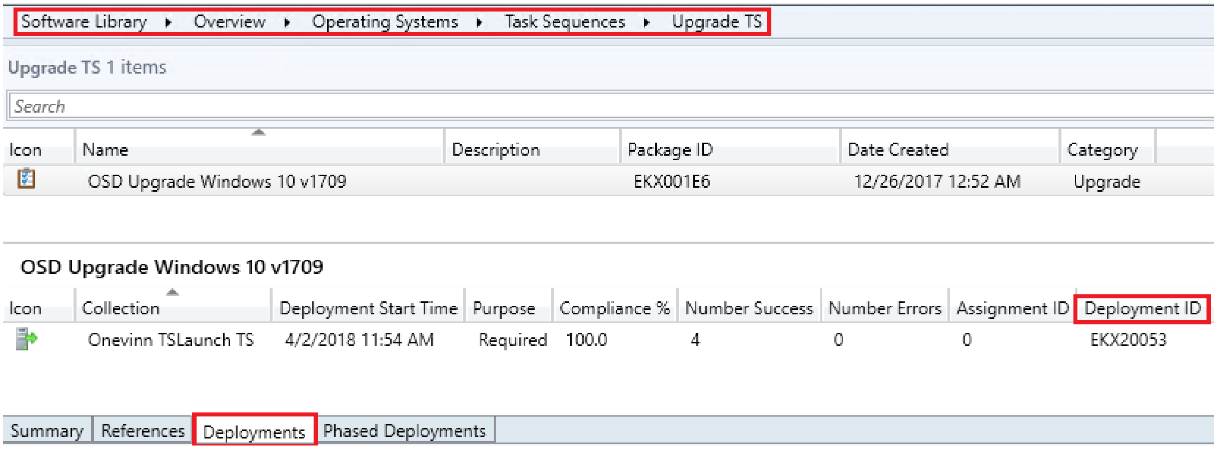
```
<?xml version="1.0"?>
<configuration>
  <appSettings>
    <add key="DeploymentId" value="XYX01234"/>
    <add key="UpgradePkgId" value="XYX01234"/>
    <add key="BuildNumber" value="17134"/>
    <add key="LogLocation" value=""/>
    <add key="LogDateTimeFormat" value="yyyy-MM-dd HH:mm:ss"/>
    <add key="SuppressCMLogEntrys" value="true"/>
    <add key="EndDate" value="2021-12-31"/>
    <add key="DefaultLaunchTime" value="HH:mm"/>
    <add key="DeploymentValidNoDays" value="10"/>
    <add key="StartCountdownOfValidDaysAfterSuccessfulPreflightCheck" value="True"/>
    <add key="DoNotRemoveLogonScheduledTaskOnLaunch" value="True"/>
    <add key="FinalMessageThreshold" value="3"/>
    <add key="MaxAllowedDevianceOnScheduledLaunch" value="20"/>
    <add key="DefaultToAutoStart" value="false"/>
    <add key="AllowAutoUpgradeIfNoOneLoggedOn" value="True"/>
    <add key="AllowAutoUpgradeIfNoOneLoggedOnOnlyIfScheduled" value="True"/>
    <add key="AutoStartUpgradeOnEndDate" value="True"/>
    <add key="AllowPostpone" value="True"/>
    <add key="AllowWakeToRun" value="True"/>
    <add key="ForceUpgradeAfterCountdown" value="False"/>
    <add key="ForceLogoffUser" value="True"/>
    <add key="LaunchUPGBackground" value="True"/>
    <add key="CountDownTime" value="36000"/>
    <add key="NaggerTime" value="7200"/>
    <add key="ToastInterval" value="60"/>
    <add key="FrameColor" value="#3366BB"/>
    <add key="MainBackgroundColor" value="#C0D6E4"/>
    <add key="InfoForeColor" value="#000000"/>
    <add key="LinkColor" value="#0645AD"/>
    <add key="ForceIEToOpenHyperlink" value="True"/>
    <add key="Width" value="720"/>
    <add key="Height" value="365"/>
    <add key="LogoAtTop" value="True"/>
    <add key="LogoAtTopHeight" value="35"/>
    <add key="CustomPSScript" value=""/>
    <add key="AbortOnProcess" value=""/>
    <add key="ConfirmCMCacheSize" value="True"/>
    <add key="ExpectedCacheSize" value="10240"/>
    <add key="CheckDiskSpace" value="True"/>
    <add key="RequireDiskSpacedMB" value="10240"/>
    <add key="CheckWiredNicConnected" value="True"/>
    <add key="CheckContentAvailable" value="True"/>
    <add key="RunAssessmentTestWhenContentAvailable" value="True"/>
    <add key="AssessmentTestArguments" value="/ImageIndex 3 /auto Upgrade /quiet /noreboot /DynamicUpdate Disable /compat ScanOnly"/>
    <add key="SkipAssessmentTestOnFinalPreflightCheck" value="True"/>
    <add key="AssessmentSucessScript" value=""/>
    <add key="AssessmentFailedScript" value=""/>
    <add key="CheckACConnected" value="True"/>
    <add key="AllowUserRemediationOfAc" value="True"/>
    <add key="AllowUserRemediationOfAcDuration" value="300"/>
    <add key="ScheduledUpgradeTodayToastDuration" value="10"/>
    <add key="CheckIfPresentationIsRunning" value="True"/>
    <add key="CheckWebEndpoint" value="False"/>
    <add key="CheckWebUrl" value="http://server.domain.com/sms_mp/.sms_aut?mplist"/>
    <add key="CheckPingServer" value="False"/>
    <add key="PingServer" value="server.domain.com"/>
    <add key="CheckRebootComponentBasedServicing" value="True"/>
    <add key="CheckRebootWindowsUpdate" value="True"/>
    <add key="CheckRebootPendingFileRenameOperations" value="False"/>
    <add key="PrelaunchScript" value=""/>
  </appSettings>
</configuration>
```

9. CONFIG FILE KEYS ONE BY ONE

You cannot remove any of these keys, TSLaunch checks the presence of each and every one when launched. If you are running in DEBUG mode a message box will tell you what's missing, if not, it will be written to the log, c:\windows\temp\TSLaunch.log.

9.1. DeploymentId

TSLaunch must know which deployment to launch, the id can be found here in the console:

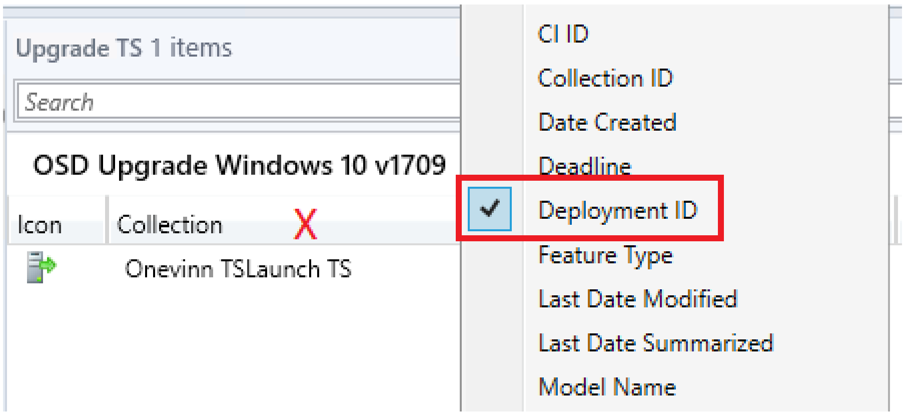

Software Library ▸ Overview ▸ Operating Systems ▸ Task Sequences ▸ Upgrade TS
Upgrade TS 1 items
Search
Summary
References
Deployments
Phased Deployments

Icon	Name	Description	Package ID	Date Created	Category
	OSD Upgrade Windows 10 v1709		EKX001E6	12/26/2017 12:52 AM	Upgrade

OSD Upgrade Windows 10 v1709

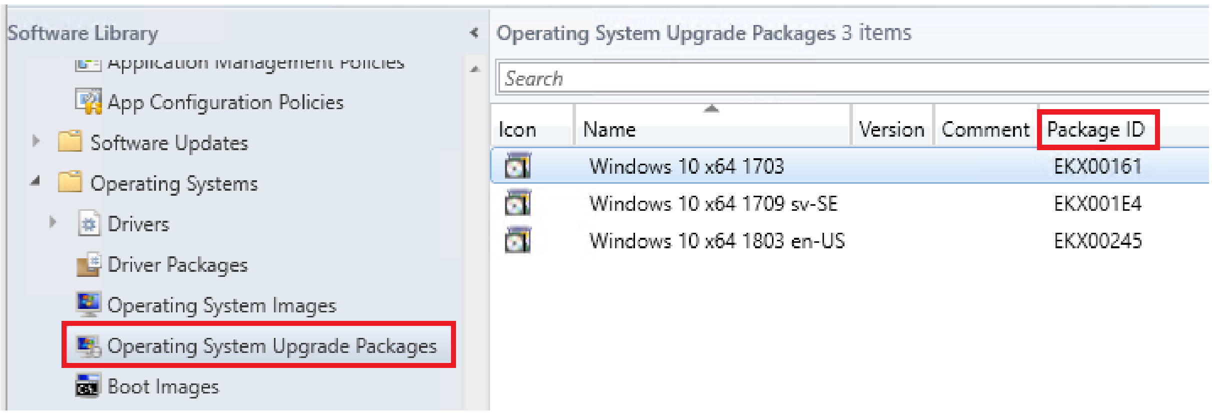
Icon	Collection	Deployment Start Time	Purpose	Compliance %	Number Success	Number Errors	Assignment ID	Deployment ID
	Onevinn TSLaunch TS	4/2/2018 11:54 AM	Required	100.0	4	0	0	EKX20053

If the column **Deployment ID** isn't visible right click the title bar and tick it:



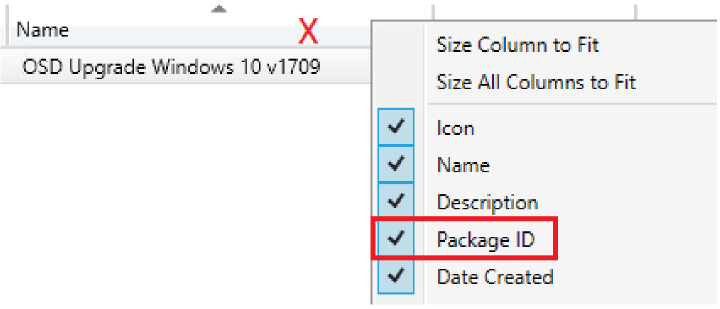
9.2. UpgradePkgId

To run an assessment test prior to launching the upgrade TSLaunch needs to know the media package id, which is found here in the console:


Software Library
Application Management Policies
App Configuration Policies
Software Updates
Operating Systems
Drivers
Driver Packages
Operating System Images
Operating System Upgrade Packages
Boot Images
Operating System Upgrade Packages 3 items
Search

Icon	Name	Version	Comment	Package ID
	Windows 10 x64 1703			EKX00161
	Windows 10 x64 1709 sv-SE			EKX001E4
	Windows 10 x64 1803 en-US			EKX00245

If the column **Package ID** isn't visible right click the title bar and tick it:



9.3. BuildNumber

The Windows 10 build number you are **upgrading to**.

Example: "**17134**"

9.4. LogLocation

Network share for logging, SYSTEM ("Domain Computers") needs modify permissions on the share.

Example: "**\\Server.domain.com\TSLaunchLogs$**" or ""

This is not mandatory, leave blank if not used. Do not remove the key.

9.5. LogDateTimeFormat

Set your preferred time format for TSLaunch logs.

Example: “**G**” or “**yyyy-MM-dd HH:mm:ss**”

9.6. SuppressCMLogEntrys

Set this key to “True” to reduce the number of status messages sent to ConfigMgr. This is useful in big environments running a high number of simultaneous upgrades.

Examples: “**True**”

9.7. EndDate

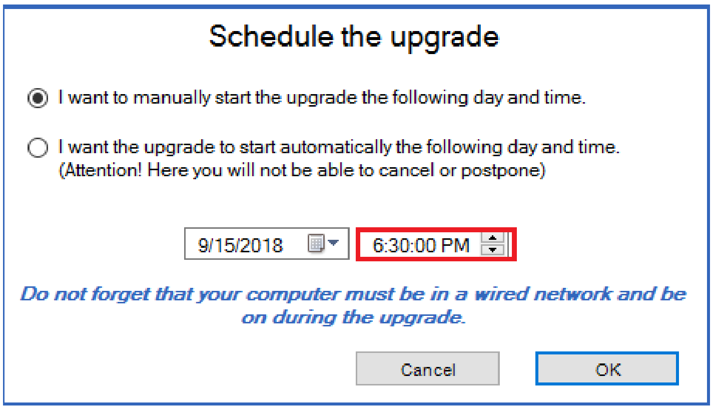
Set the end date for you roll out. If your honoring the concept of “rings” this will be just after your final ring.

Example: “**2019-01-20**” or “**01/20/2019**”

9.8. DefaultLaunchTime

Default time pre-choosen in TSLaunch scheduler window.

Example: “**18:30**” – use **24-hour format** for this.



Schedule the upgrade

● I want to manually start the upgrade the following day and time.

○ I want the upgrade to start automatically the following day and time.
(Attention! Here you will not be able to cancel or postpone)

9/15/2018 6:30:00 PM

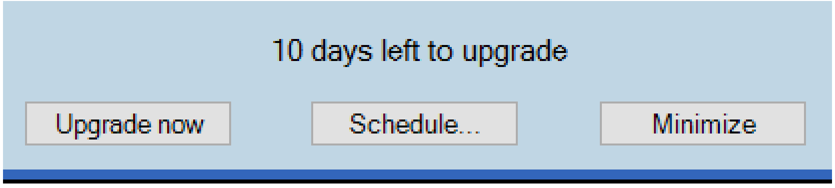
Do not forget that your computer must be in a wired network and be on during the upgrade.

9.9. DeploymentValidNoDays

To enable reuse of the package for the duration of the upgrade (all rings) each ring can be set to valid for a certain number of days in this key.

Example: “**10**”

The number of days left is counted down and presented in the UI.



10 days left to upgrade

9.10. StartCountdownOfValidDaysAfterSuccessfulPreflightCheck

Refers to the previous key “DeploymentValidNoDays” and determines if the countdown of days should start when TSLaunch is first run or after the first successful preflight check. Many prefer the latter as TSLaunch never interacts with the user until successfully completing all checks.

Example: “**True**” or “**False**”

9.11. DoNotRemoveLogonScheduledTaskOnLaunch

TSLaunch is run by the ConfigMgr client only once, all consecutive launches of the application is handled by two scheduled tasks. To assure that the attempts to upgrade continues even after an upgrade failure (such as rollback) you should set this key to “True”. This will result in TSLaunch running one final time after the upgrade has completed. If the upgrade succeeded TSLaunch will detect that the Operating system is already upgraded and exit silently, otherwise it will give the user an opportunity to reschedule.

Example: “**True**” or “**False**”

9.12. FinalMessageThreshold

Days left when to switch to "stronger" finalmessage.rtf

The language folder(s) should always contain two rtf files “message.rtf” and “finalmessage.rtf”. When the specified number of days remain TSLaunch will switch to the presumably stronger message in finalmessage.rtf, trying to convince the user to run or at least schedule the upgrade.

This behavior cannot be disabled, if you don’t wish to use it, just put the same message in both files.

Example: “**3**”

9.13. MaxAllowedDevianceOnScheduledLaunch

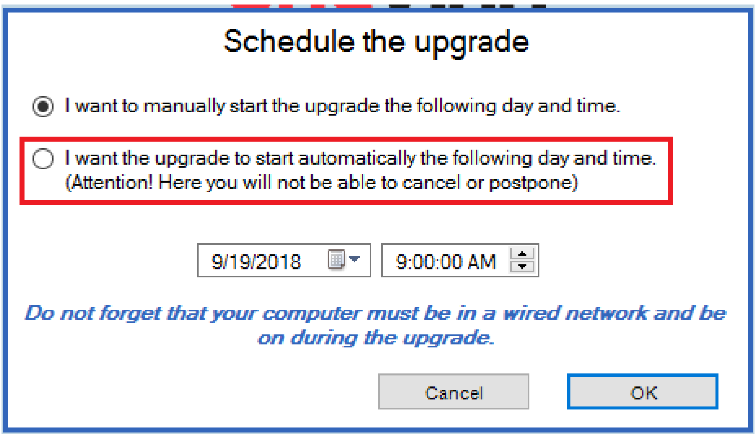
This value (in minutes) specifies the max allowed deviance from a user scheduled upgrade when TSLaunch will still launch the upgrade TS.

If a user schedules the upgrade to, for example, 07:00 PM, turns off the machine and goes home, it would cause irritation if the upgrade was launched at 9 o'clock the morning after, when he or she logs back on and really needs the machine. If set to for example "20" the upgrade will be suppressed after 07:20 PM and the user is asked to reschedule.

Example: "**20**"

9.14. DefaultToAutoStart

Setting this key to "True" will switch the pre-chosen alternative in the scheduler dialog to auto start the upgrade when the scheduled time occurs.



Schedule the upgrade

● I want to manually start the upgrade the following day and time.

○ I want the upgrade to start automatically the following day and time. (Attention! Here you will not be able to cancel or postpone)

9/19/2018 9:00:00 AM

Do not forget that your computer must be in a wired network and be on during the upgrade.

Example: "**True**" or "**False**"

9.15. AllowAutoUpgradeIfNoOneLoggedOn

If set to "True" the upgrade TS will, if no one is logged on, be launched automatically as soon as all preflight checks are successful. This can be restricted in the next variable.

Example: "**True**" or "**False**"

9.16. AllowAutoUpgradeIfNoOneLoggedOnOnlyIfScheduled

Setting this key to "True" restricts "auto upgrade if no one is logged on" to be allowed only if TSLaunch is run on a user defined schedule. This is useful if one wish to minimize surprises but still allow computers to be upgraded when the user is absent.

Example: "**True**" or "**False**"

9.17. AutoStartUpgradeOnEndDate

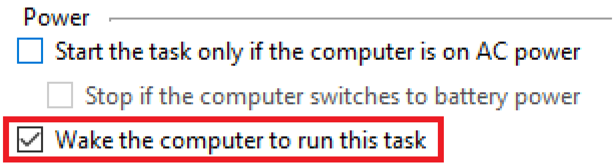
This key controls whether TSLaunch automatically should launch the upgrade TS when the End Date is reached. Doing so must be considered as a last resort to force the upgrade on computers where the user has neglected all attempts to convince him to schedule it freely.

If used (True) it might be a good idea to notify the user that this is going to happen with a few words in the finalmessage.rtf.

Example: “**True**” or “**False**”

9.18. AllowWakeToRun

Apart from the initial launch, which is done by the ConfigMgr client, TSLaunch is triggered by scheduled tasks. Scheduled tasks can at some circumstances wake the computer from sleep or hibernation. Setting this key to “True” will instruct TSLaunch to create and use tasks with this setting enabled.



Power

□ Start the task only if the computer is on AC power

□ Stop if the computer switches to battery power

☑ Wake the computer to run this task

Example: “**True**” or “**False**”

9.19. ForceUpgradeAfterCountdown

Whenever TSLaunch is started it should be considered a “session” under which we try to establish contact with and persuade the user to schedule the upgrade. The length of such a session before giving up is set in the key “CountDownTime” further down.

Normally it’s probably not desired but it is possible to have TSLaunch forcefully start the upgrade at the end of a session.

Example: “**True**” or “**False**”

9.20. ForceLogoffUser

Controls whether to automatically logoff the user when the upgrade starts.

Example: “**True**” or “**False**”

9.21. LaunchUPGBackground

TSLaunch has the capability to start the full screen application UPGBackground, described in a separate section in this document, when the upgrade TS is launched.

Example: “**True**” or “**False**”

9.22. CountDownTime

The length of a “session” in seconds. Normally this should be set to just over a work day. It must never exceed 23 hrs.

Example: “**36000**” (10 hrs.)

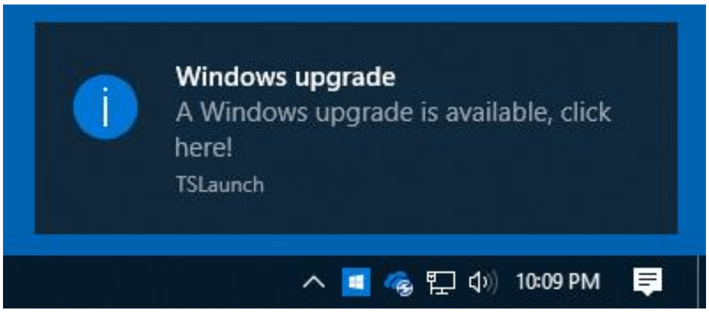
9.23. NaggerTime

This setting (seconds) control how often a minimized TSLaunch will be restored back. When restored TSLaunch will appear mid screen as topmost window, reminding the user to schedule or upgrade.

Example: “**7200**” (2 hrs.)

9.24. ToastInterval

Whilst TSLaunch is minimized it’s possible to have it show a “balloon” at certain intervals. This setting (seconds) tells TSLaunch how often. Set to a very high value to omit it, like “999999”



Windows upgrade

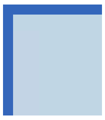
A Windows upgrade is available, click here!

TSLaunch

Example: “**1800**” (30 minutes)

9.25. FrameColor

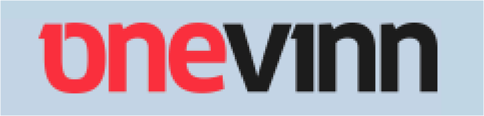
Defines the outer frame color of the main window.



Example: “**#3366BB**”

9.26. MainBackgroundColor

Defines the background color of the main window.

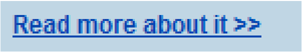


onevinn

Example: “**#C0D6E4**”

9.27. LinkColor

Defines in which color the Hyperlink text is presented. The URL behind can be found in the UILanguage.xml file for each supported language.



Read more about it >>

Example: “**#0645AD**”

9.28. ForceIEToOpenHyperlink

Set key to “True” to force Hyperlink to open in IE (if enterprise mode is used it otherwise seem to cause a very long delay)

Example: “**True**” or “**False**”

9.29. Width

The width of the main window; adjust to accommodate your text (message.rtf).

Example: “**720**”

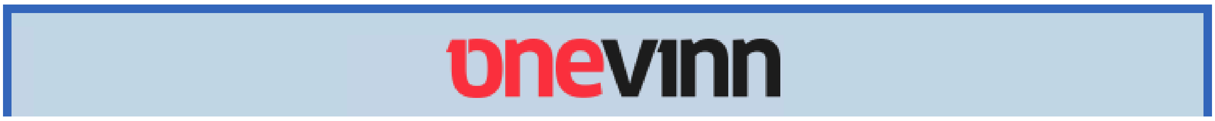
9.30. Height

The height of the main window; adjust to accommodate your text (message.rtf).

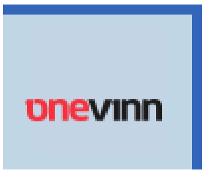
Example: “**365**”

9.31. LogoAtTop

If set to “True” the TSLaunch.png (logo) will be placed at top of the main window.



The Onevinn logo is not at all suited for it but should you have a square shaped logo it’s probably better placed at the right side of the window (False). These are the two available options.



Example: “**True**” or “**False**”

9.32. LogoAtTopHeight

When the logo is placed at top, one can adjust the height (pixels) of the picture box holding it. This is useful to get an as nice-looking design as possible.

Example: “**35**”

9.33. CustomPSScript

TSLaunch has several built-in preflight checks, each of them can be turned off and on by True/False switches. They will be covered further down.

At some occasions the built-ins are just not enough, in those cases you can let TSLaunch run your own custom script, an example “CheckMcAfeeVersion.ps1” has been included, study it for better understanding and remember to test thoroughly before deploying. **Your script will be executed as SYSTEM.**

The script must exit with return code 0 on success, all other values will be considered a failure.

Example: “**MyCustomPreflightCheck.ps1**”

9.34. AbortOnProcess

This is the first of the built-in preflight checks, the ability to check if one or several processes are running and postpone interaction with the user until not.

Example: “**Notepad:MSPaint**”

9.35. ConfirmCMCacheSize

Whether TSLaunch should check the cache size – this will not cancel the upgrade, only log a warning.

Example: “**True**” or “**False**”

9.36. ExpectedCacheSize

The expected size of CMCache.

Example: “**20240**”

9.37. CheckDiskSpace

If TSLaunch should abort (fatal) if not enough free HDD space is available.

Example: “**True**” or “**False**”

9.38. RequireDiskSpacedMB

The required HDD space in MB.

Example: “**10240**”

9.39. CheckWiredNicConnected

If TSLaunch should check for wired network connection.

Example: “**True**” or “**False**”

9.40. CheckContentAvailable

This key must only be set to “True” if the task sequence deployment is also configured to pre-download all content before running. Setting it to true will tell TSLaunch to, as part of the preflight checks, secure that all content, Upgrade media and packages, are present in the CCMCACHE before interacting with the user. It is also crucial that the upgrade package id (UpgradePkgId) has been correctly filled in above to make this feature work.

Example: “**True**” or “**False**”

9.41. RunAssessmentTestWhenContentAvailable

If the task sequence deployment is configured to pre-download all content before running it’s possible to have TSLaunch perform an “Assessment test”. This will be performed according to the settings in the next key “AssessmentTestArguments”.

The result of the test will be logged as integer, hex and, in most cases, clear text.

Example: “**True**” or “**False**”

Remark: Running an Assignment test might temporarily impose a slight negative impact on the computer’s performance. Since TSLaunch is running repeated preflight checks until all pass the assessment test will be suppressed for 12 hours. once succeeded. This will be logged.

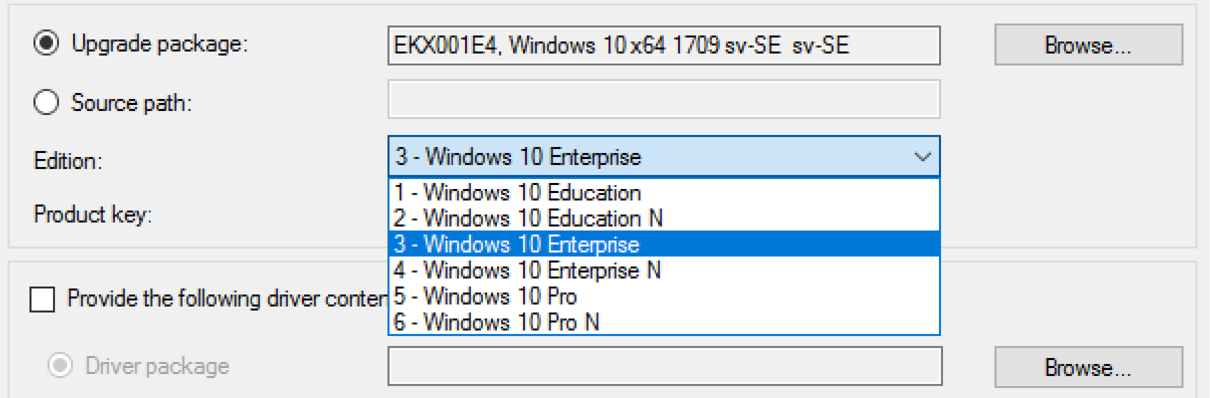
AssessmentTestArguments

When an assessment test is performed during (within) an upgrade task sequence it invokes the windows installer program “Setup.exe” with the following arguments:

“/ImageIndex 3 /auto Upgrade /quiet /noreboot /DynamicUpdate Disable /compat ScanOnly”

The “AssementTestArguments” key is prepopulated with the same switches and arguments.

The image index might have to be changed if you’re upgrading to anything apart from the Enterprise edition, are using modified (patched) media or if Microsoft in the future decides to not bundle the different editions. For 1709 and 1803 out of the box media, no change is required.



● Upgrade package: EKX001E4, Windows 10 x64 1709 sv-SE sv-SE

○ Source path:

Edition: 3 - Windows 10 Enterprise

Product key:

1 - Windows 10 Education
2 - Windows 10 Education N
3 - Windows 10 Enterprise
4 - Windows 10 Enterprise N
5 - Windows 10 Pro
6 - Windows 10 Pro N

□ Provide the following driver conter

○ Driver package

It’s possible to add an extra switch/argument to the line if you wish to have the logs copied somewhere for further analysis.

/CopyLogs <folder_path>\%COMPUTERNAME%

%COMPUTERNAME% will in this case be resolved by TSLaunch.

9.42. SkipAssessmentTestOnFinalPreflightCheck

Even if the preflight checks have previously succeeded TSLaunch will always perform a final check just before launching the upgrade task sequence. This is necessary to detect if, for example, the user has disconnected the AC power or left the company network since last run.

It might not be necessary to run an assessment test as part of this final check; thus, it can be suppressed by setting this key to “True”.

Example: “**True**” or “**False**”

9.43. AssessmentSucessScript

There are many possible outcomes of an assessment test, one is success all the others are different kinds of failures, all preventing the upgrade.

If the test is successful you might want to take some special action, like adding the computer to a collection or similar. To accomplish this, it’s possible to run a PowerShell script on success. The script could in turn invoke a web service, such as OnevinnWS, to do the actual adding. An example script that does just that has been included. **Executed as SYSTEM.**

Leave blank if no script should be run.

Example: “**MyOnAssessmentSuccessScript.ps1**” or “”

9.44. AssessmentFailedScript

Same as above but the script will be invoked in the event of a failure. **Executed as SYSTEM**

Example: “**MyOnAssessmentFailureScript.ps1**” or “”

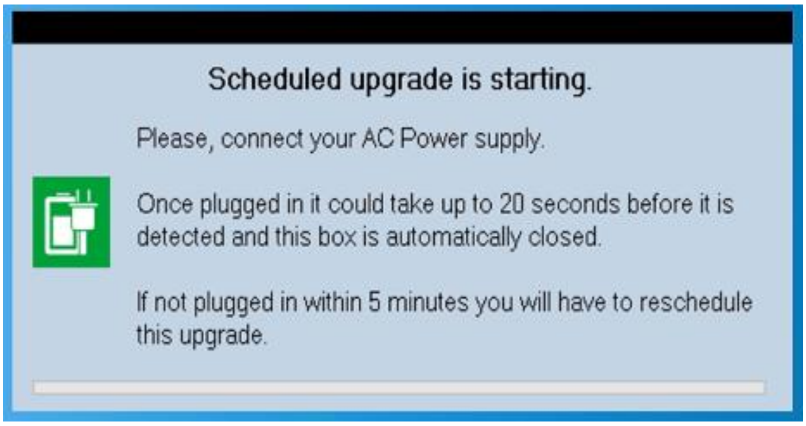
9.45. CheckACConnected

Set to “True” to have TSLaunch check that the computer runs on AC power.

Example: “**True**” or “**False**”

9.46. AllowUserRemediationOfAc

If “True” the following dialog will appear encourage the user to plug-in AC Power supply, if not already connected.



Scheduled upgrade is starting.

Please, connect your AC Power supply.

Once plugged in it could take up to 20 seconds before it is detected and this box is automatically closed.

If not plugged in within 5 minutes you will have to reschedule this upgrade.

The text can be modified as necessary in << UILanguage.xml >>.

Example: “**True**” or “**False**”

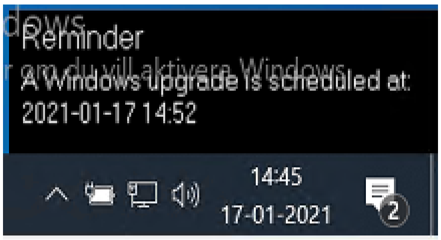
9.47. AllowUserRemediationOfAcDuration

Number of seconds the user is granted to plug in AC Power.

Example: “**300**” (5 minutes)

9.48. ScheduledUpgradeTodayToastDuration

If set to anything but “0” TSLaunch will remind the user about the upcoming upgrade. This reminder will be shown twice on the day of schedule; 15 minutes after logon or at 10 AM, and 5 minutes before the scheduled time.



Reminder

A Windows upgrade is scheduled at:
2021-01-17 14:52

(sorry for the poor picture)
Example: “5” (Toast will be visible for 5 seconds)

9.49. CheckIfPresentationIsRunning

To make sure a user isn't disturbed during a presentation (PowerPoint) TSLaunch can detect it. Set to "True" to activate the check.

Example: "**True**" or "**False**"

9.50. CheckWebEndpoint

When "True" check the availability of the web endpoint (Url) in the next key. This is one of two options to make sure the computer is on a certain network or network segment.

Example: "**True**" or "**False**"

9.51. CheckWebUrl

The web endpoint (Url) to check. In the example insert the name of a Management point server and that will be used.

Example: "**http://<server.domain.com>/sms_mp/.sms_aut?mplist**"

9.52. CheckPingServer

This is the second option to determine that the computer is on the correct location. Set to "True" to turn on, TSLaunch will then try to Ping the server in the next key as part of the preflight checks.

Example: "**True**" or "**False**"

9.53. PingServer

The name of the server to ping, use FQDN.

Example: "**server.domain.com"**

9.54. PrelaunchScript

The last script hook runs a script just prior to launching the upgrade TS. **Executed as SYSTEM**

Example: "**MyPrelaunchScript.ps1**"

10. DEPLOYMENTS

10.1. Thumb rules

#1: If you wish to make the deployment visible in Software center, create an "Available" deployment. This does not require any special scheduling.

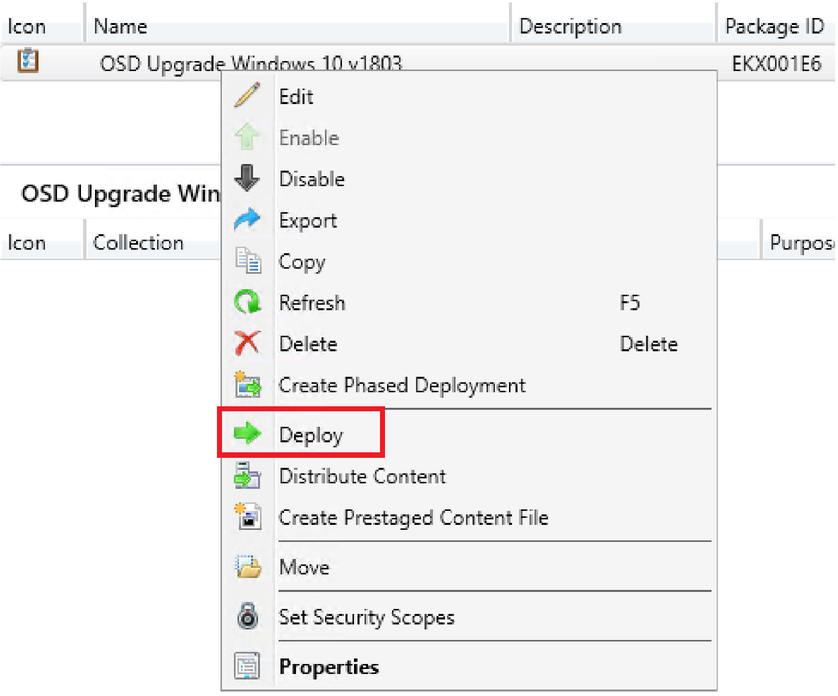
#2: If you wish to completely hide you task sequence from your users, create a "Required" deployment. This has to be scheduled in a way that only TSLaunch will engage it.

#3: If you intend to run an Assessment test prior to launching the TS and/or be able to run the upgrade off site, you will need to use "Pre-download content for this task sequence" in combination with "Download all content locally before starting task sequence". This rule applies to both available and required.

10.2. Task sequence

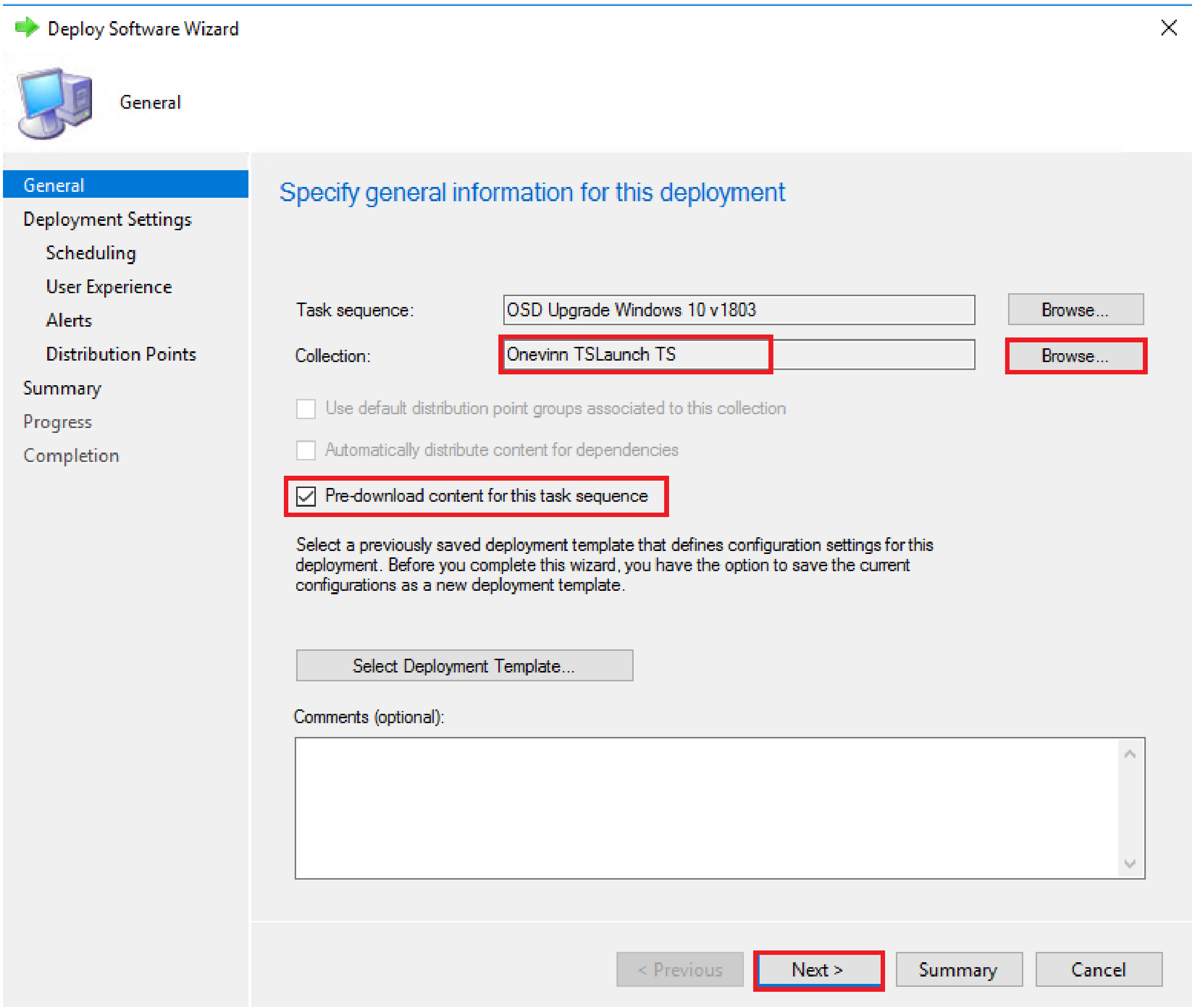
The scenario that has generated far most questions on Technet gallery concerns downloaded content, so we will cover that in detail.

1. Right click your TS and click Deploy



Icon	Name	Description	Package ID
	OSD Upgrade Windows 10 v1803		EKX001E6

2. Browse for and choose you deployment collection. Make sure to tick “**Pre-download content for this task sequence**”. Click Next.



Deploy Software Wizard

General

Specify general information for this deployment

Task sequence: OSD Upgrade Windows 10 v1803

Collection: Onevinn TSLaunch TS

□ Use default distribution point groups associated to this collection

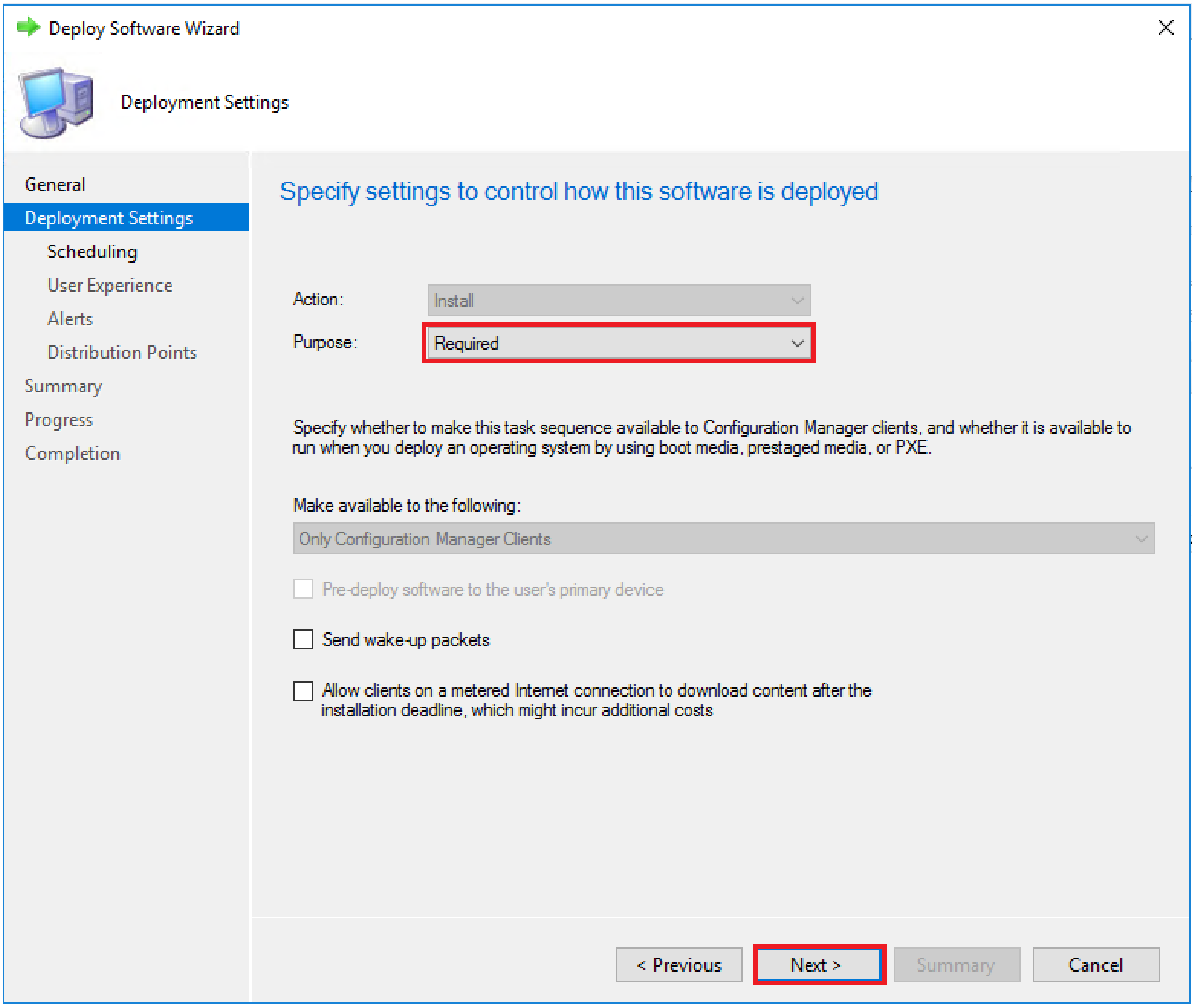
□ Automatically distribute content for dependencies

☑ Pre-download content for this task sequence

Select a previously saved deployment template that defines configuration settings for this deployment. Before you complete this wizard, you have the option to save the current configurations as a new deployment template.

Comments (optional):

3. Make the deployment “**Required**”. Click Next.



Deploy Software Wizard

Deployment Settings

Specify settings to control how this software is deployed

Action: Install

Purpose: Required

Specify whether to make this task sequence available to Configuration Manager clients, and whether it is available to run when you deploy an operating system by using boot media, prestaged media, or PXE.

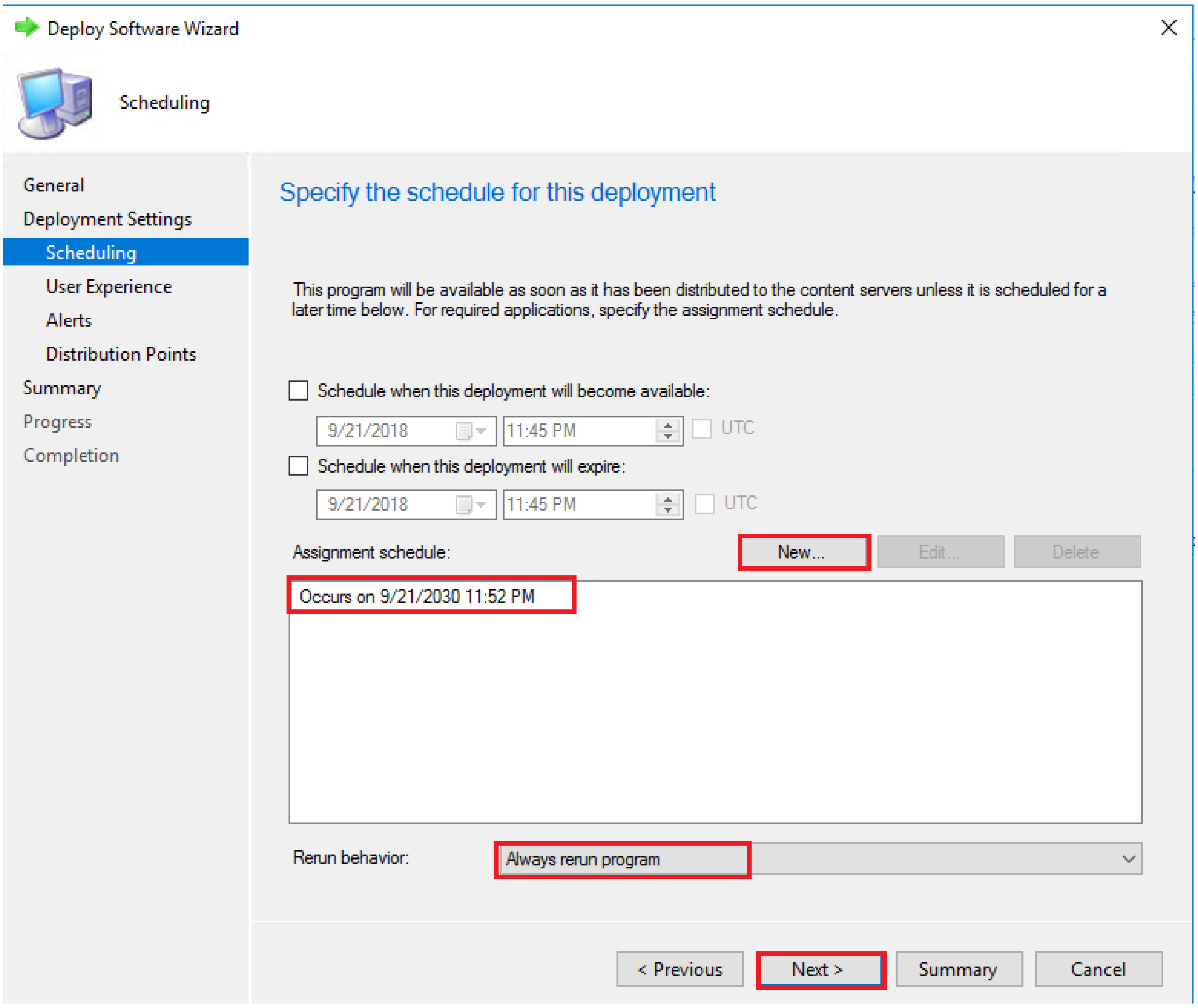
Make available to the following: Only Configuration Manager Clients

□ Pre-deploy software to the user's primary device

□ Send wake-up packets

□ Allow clients on a metered Internet connection to download content after the installation deadline, which might incur additional costs

4. Press New and add a schedule far into the future (2030). Choose Run behavior. Click Next.



Specify the schedule for this deployment

This program will be available as soon as it has been distributed to the content servers unless it is scheduled for a later time below. For required applications, specify the assignment schedule.

☐ Schedule when this deployment will become available: 9/21/2018 11:45 PM ☐ UTC

☐ Schedule when this deployment will expire: 9/21/2018 11:45 PM ☐ UTC

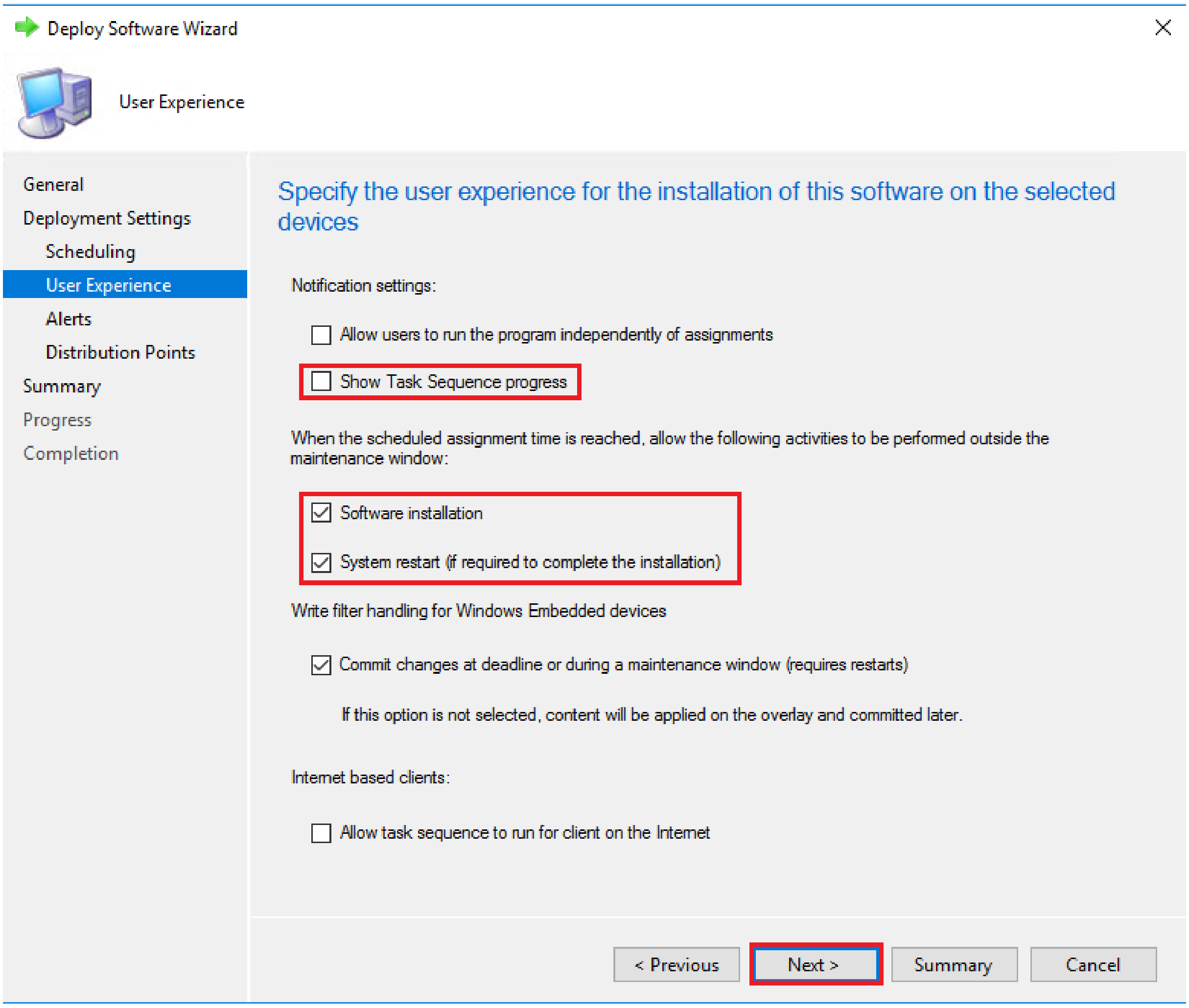
Assignment schedule:

Occurs on 9/21/2030 11:52 PM

Rerun behavior: Always rerun program

Note the Year = **2030**

5. Make three changes on the User Experience tab. **Click Next twice**.



Deploy Software Wizard

User Experience

General
Deployment Settings
Scheduling
User Experience
Alerts
Distribution Points
Summary
Progress
Completion

Specify the user experience for the installation of this software on the selected devices

Notification settings:

□ Allow users to run the program independently of assignments

□ Show Task Sequence progress

When the scheduled assignment time is reached, allow the following activities to be performed outside the maintenance window:

☑ Software installation

☑ System restart (if required to complete the installation)

Write filter handling for Windows Embedded devices

☑ Commit changes at deadline or during a maintenance window (requires restarts)

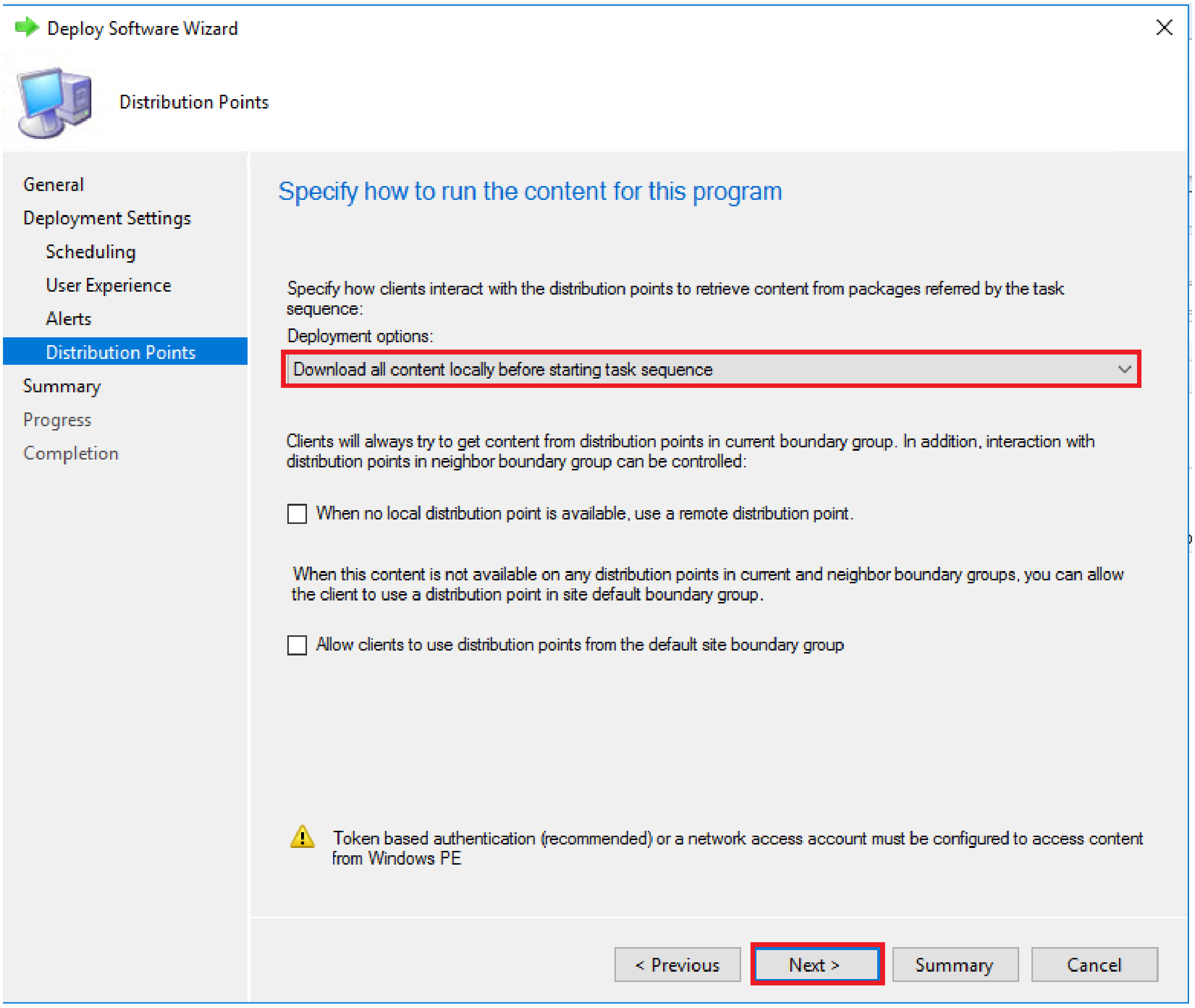
If this option is not selected, content will be applied on the overlay and committed later.

Internet based clients:

□ Allow task sequence to run for client on the Internet

Remark: Since TSLaunch allows the user to schedule the upgrade freely we need to make sure the deployment can run outside of maintenance windows. So, we must check the two boxes in the middle.

6. Choose option “Download all content locally before starting…..”. Click Next.



Deploy Software Wizard

Distribution Points

Specify how to run the content for this program

Specify how clients interact with the distribution points to retrieve content from packages referred by the task sequence:

Deployment options: Download all content locally before starting task sequence

Clients will always try to get content from distribution points in current boundary group. In addition, interaction with distribution points in neighbor boundary group can be controlled:

☐ When no local distribution point is available, use a remote distribution point.

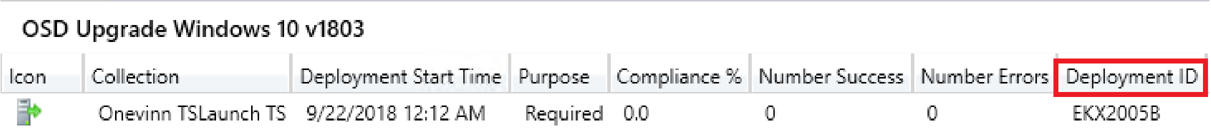
When this content is not available on any distribution points in current and neighbor boundary groups, you can allow the client to use a distribution point in site default boundary group.

☐ Allow clients to use distribution points from the default site boundary group

Token based authentication (recommended) or a network access account must be configured to access content from Windows PE

7. Click Next twice.
8. Click Close.

This is a good time to pick up the Deployment Id for the config file:



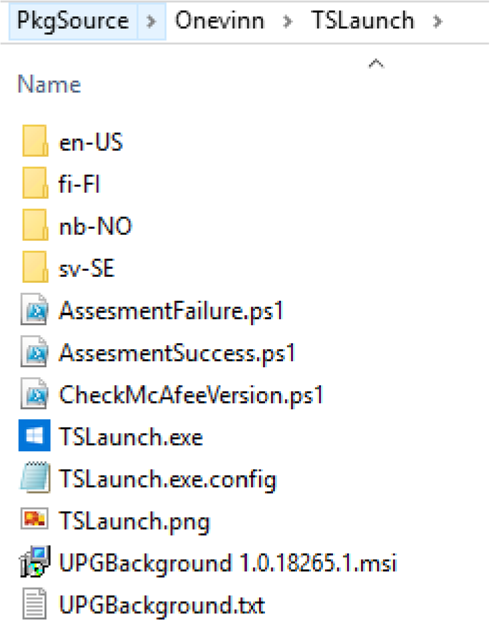
OSD Upgrade Windows 10 v1803

Icon	Collection	Deployment Start Time	Purpose	Compliance %	Number Success	Number Errors	Deployment ID
	Onevinn TSLaunch TS	9/22/2018 12:12 AM	Required	0.0	0	0	EKX2005B

```
<add key="DeploymentId" value="XYZ01234"/>
```

10.3. TSLaunch – create program

In this case we have extended the package with two additional languages and added the UPGBackgrond msi, as we have configured TSLaunch to install it.



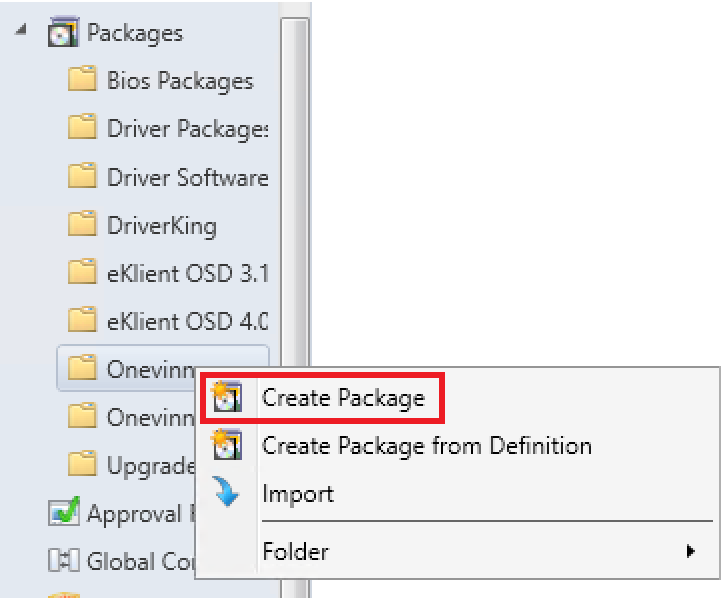
PkgSource > Onevinn > TSLaunch >

Name

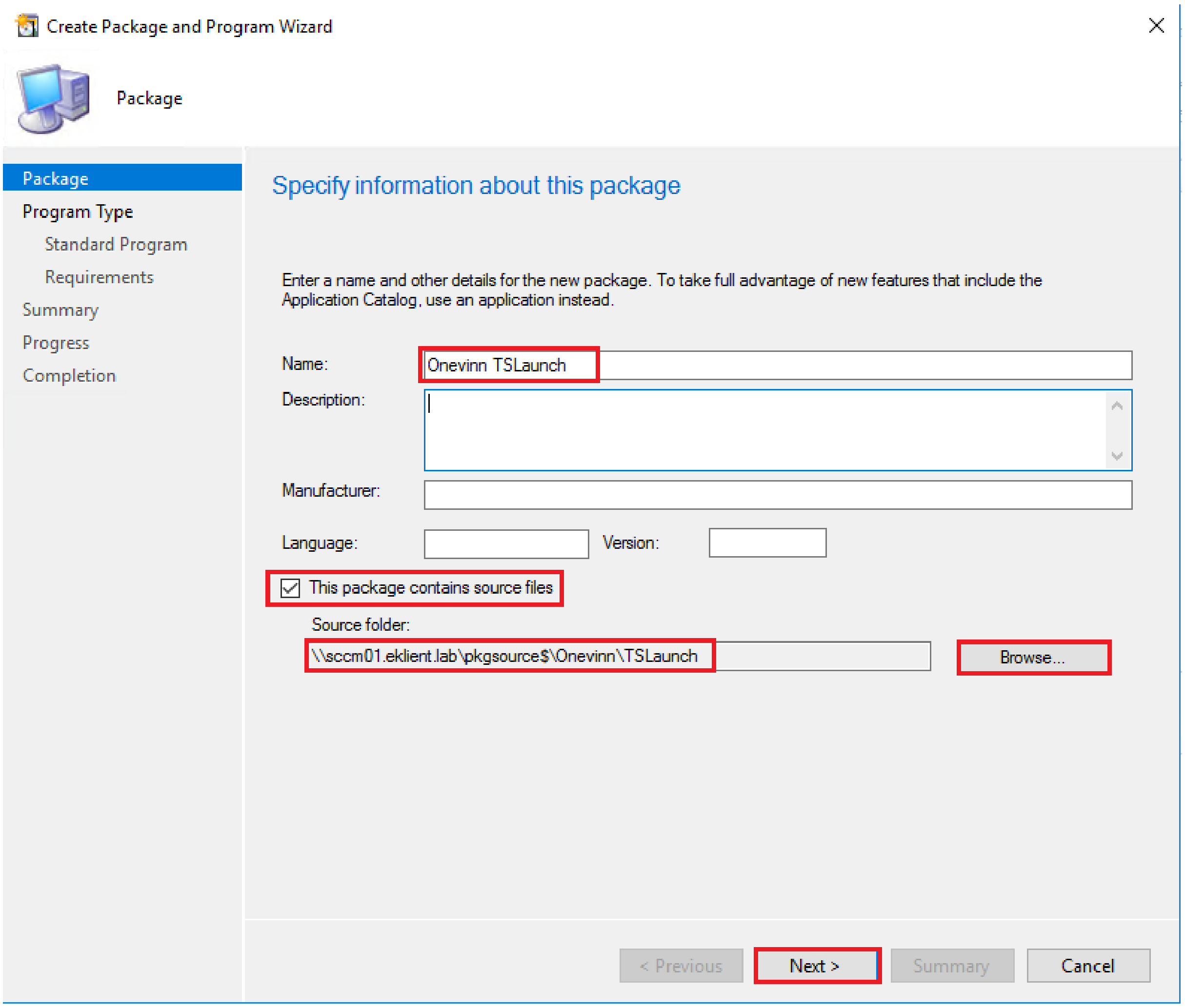
- en-US
- fi-FI
- nb-NO
- sv-SE
- AssesmentFailure.ps1
- AssesmentSuccess.ps1
- CheckMcAfeeVersion.ps1
- TSLaunch.exe
- TSLaunch.exe.config
- TSLaunch.png
- UPGBackground 1.0.18265.1.msi
- UPGBackground.txt

We will create a package with one standard program and deploy it to the same collection as previously the upgrade task sequence.

1. Browse to your preferred tools folder under packages, right click it and choose to “Create Package”.



2. Give the new package a meaningful name, check the checkbox “The package contains source files” and browse for your source folder. Click Next.



Create Package and Program Wizard

Package

Specify information about this package

Enter a name and other details for the new package. To take full advantage of new features that include the Application Catalog, use an application instead.

Name: Onevinn TSLaunch

Description:

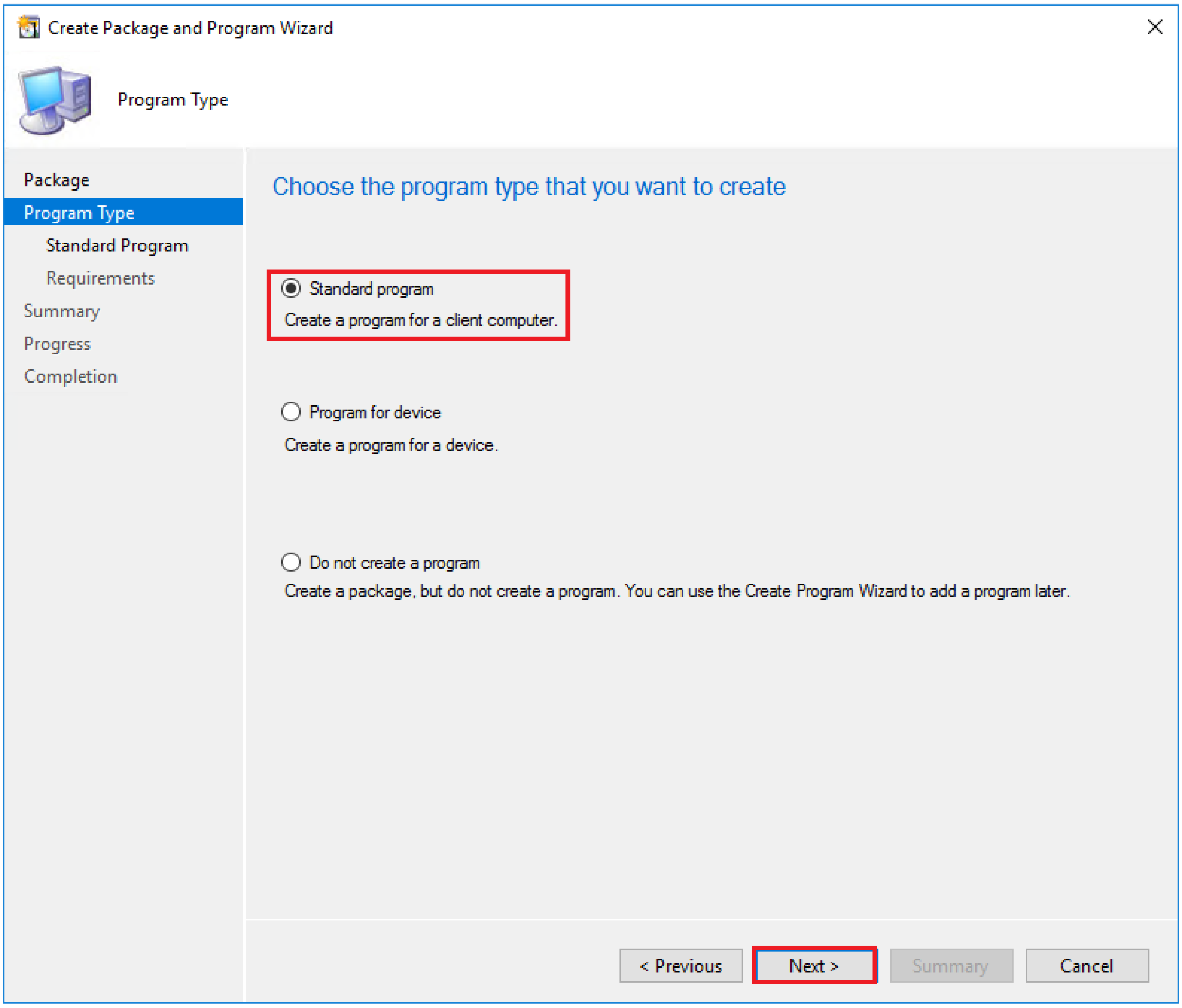
Manufacturer:

Language: Version:

☑ This package contains source files

Source folder: \\sccm01.eklient.lab\pkgsource$\Onevinn\TSLaunch

3. Choose “Standard program”. Click Next!



Create Package and Program Wizard

Program Type

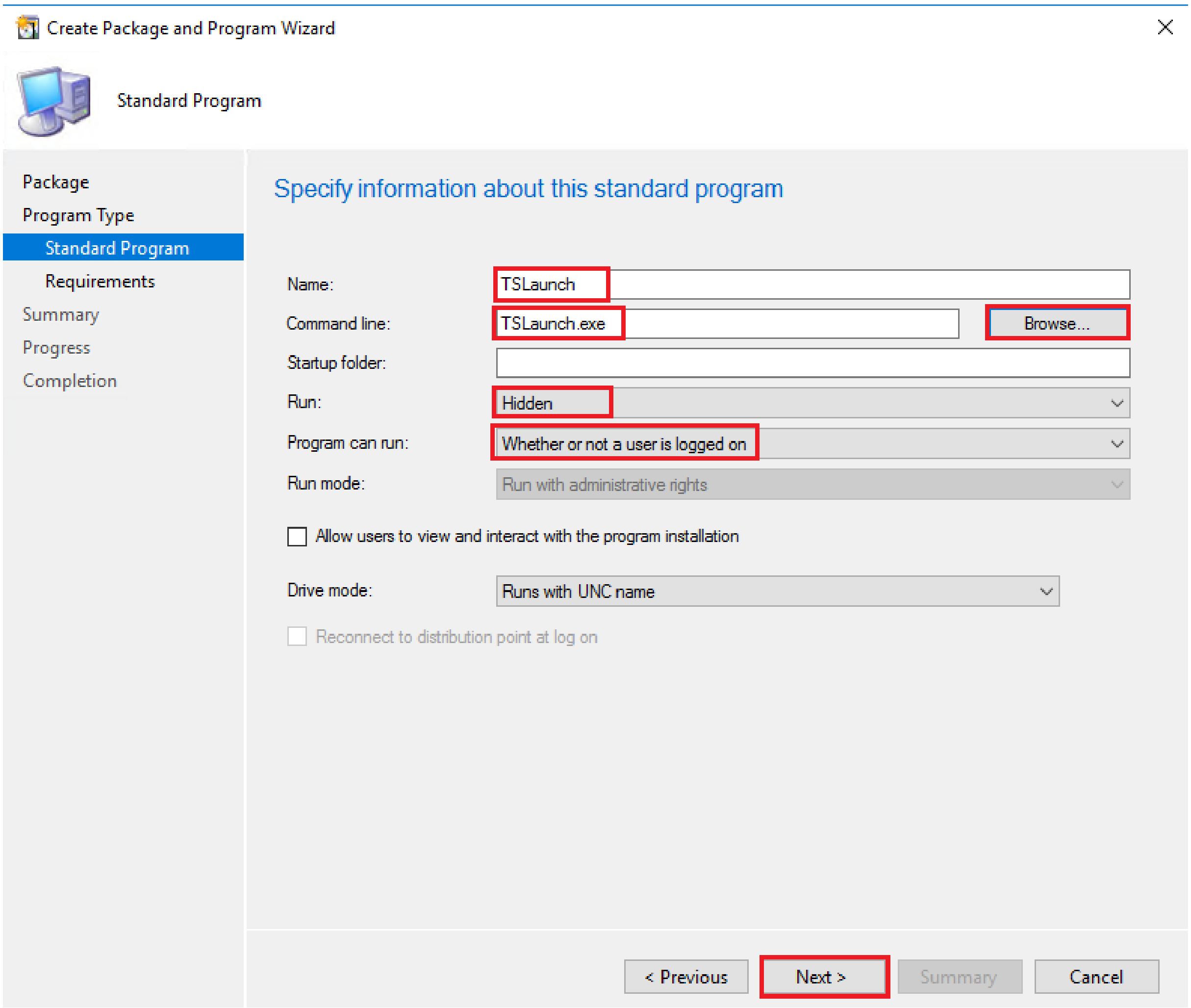
Choose the program type that you want to create

● Standard program
Create a program for a client computer.

○ Program for device
Create a program for a device.

○ Do not create a program
Create a package, but do not create a program. You can use the Create Program Wizard to add a program later.

4. Make sure to configure the program according to this picture. Click Next.



Create Package and Program Wizard

Standard Program

Specify information about this standard program

Name: TSLaunch

Command line: TSLaunch.exe

Startup folder:

Run: Hidden

Program can run: Whether or not a user is logged on

Run mode: Run with administrative rights

☐ Allow users to view and interact with the program installation

Drive mode: Runs with UNC name

☐ Reconnect to distribution point at log on

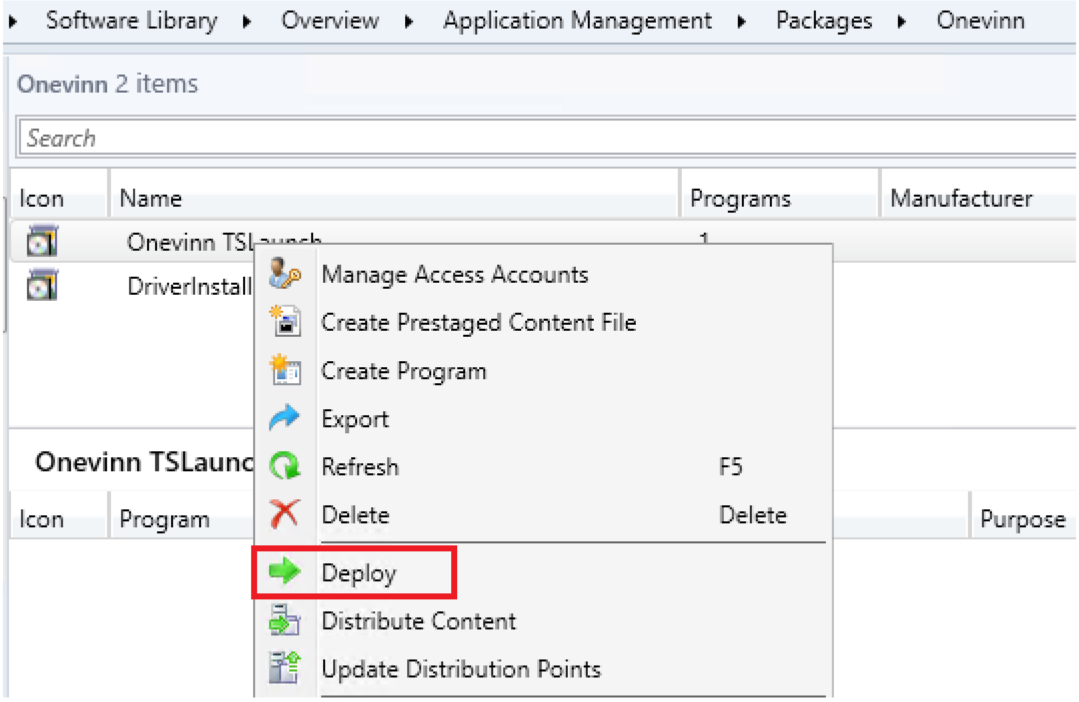
Remark: TSLaunch is designed to **run strictly in System context**. Do NOT check the box allowing user interaction, however tempting.

5. Click Next to skip the Requirements tab.
6. Click Next twice.
7. Click Close.
8. Distribute the new package to relevant Distribution points.

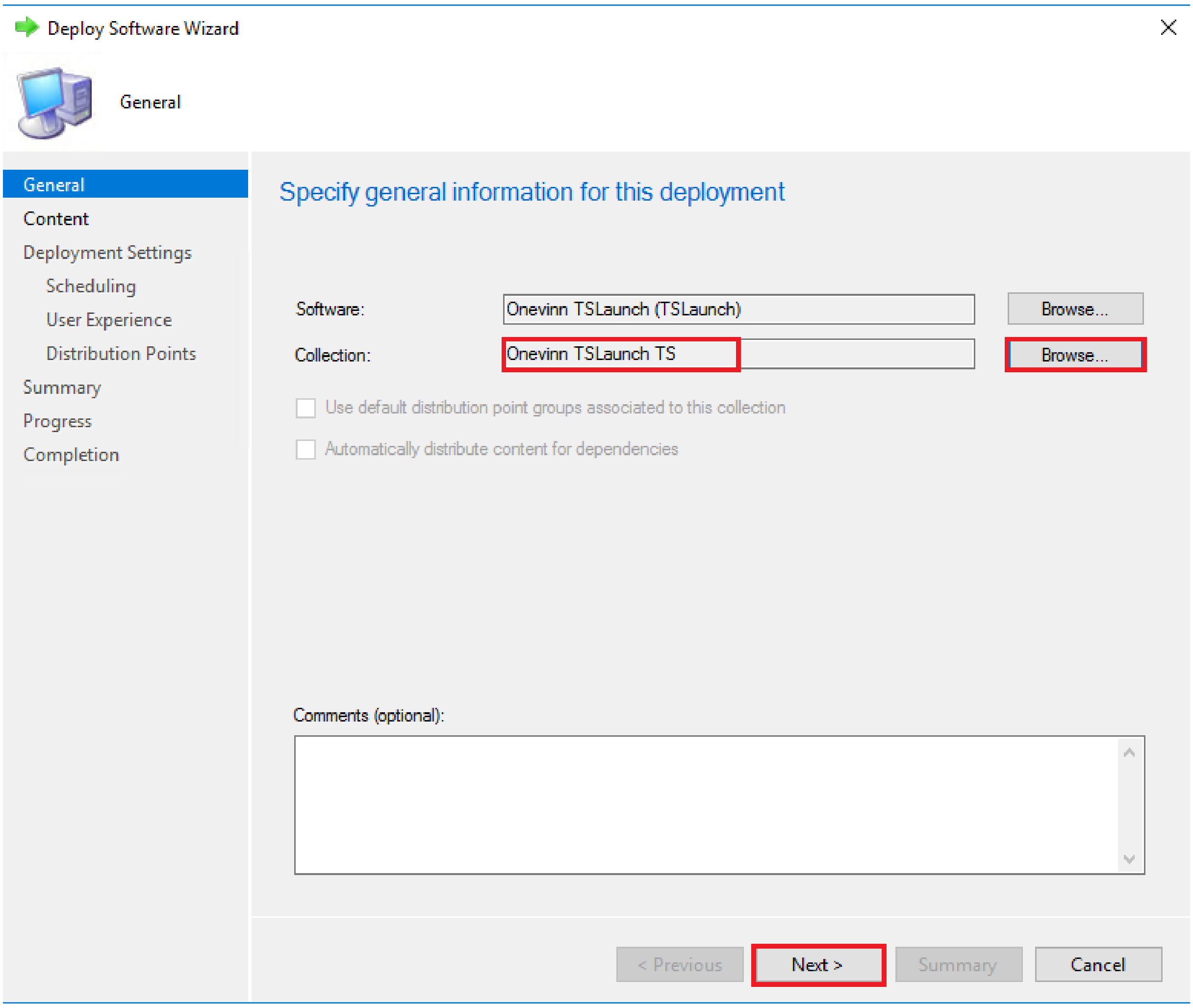
10.4. TSLaunch – deploy

In this example we will deploy TSLaunch to the same collection as previously the upgrade task sequence. This is by no means necessary, as long as it “hits” the same computers; if you have specific preferences, follow them.

1. Right click the new package and choose “Deploy”.



2. Click browse and pick your upgrade collection. Click Next.



Specify general information for this deployment

Software: Onevinn TSLaunch (TSLaunch)

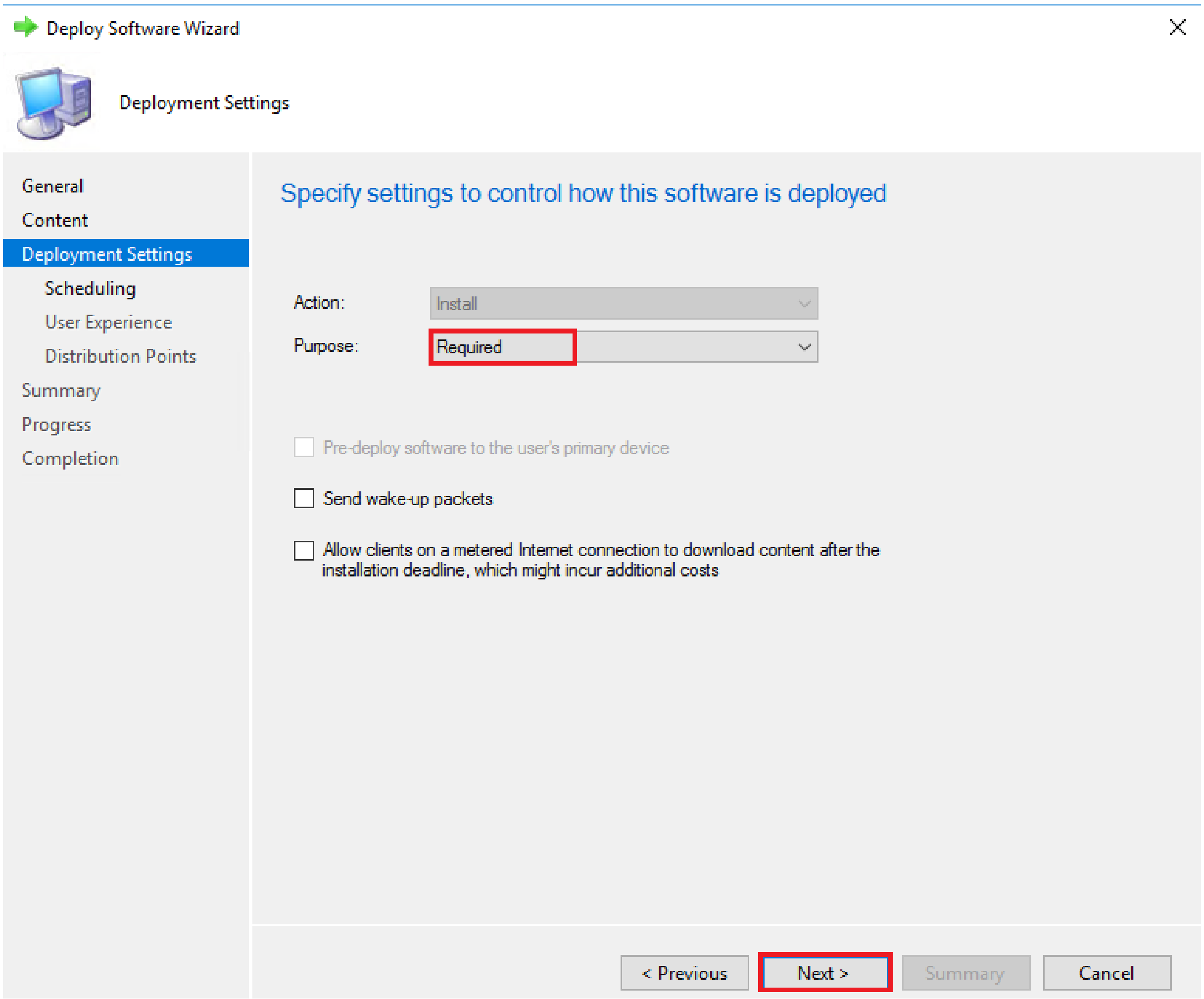
Collection: Onevinn TSLaunch TS

□ Use default distribution point groups associated to this collection

□ Automatically distribute content for dependencies

Comments (optional):

3. Click Next to skip the content tab, if you followed the instructions to create the package/program the content should already be distributed.
4. Make sure the purpose “**Required**” is picked (default). Click Next.



Deploy Software Wizard

Deployment Settings

Specify settings to control how this software is deployed

Action: Install

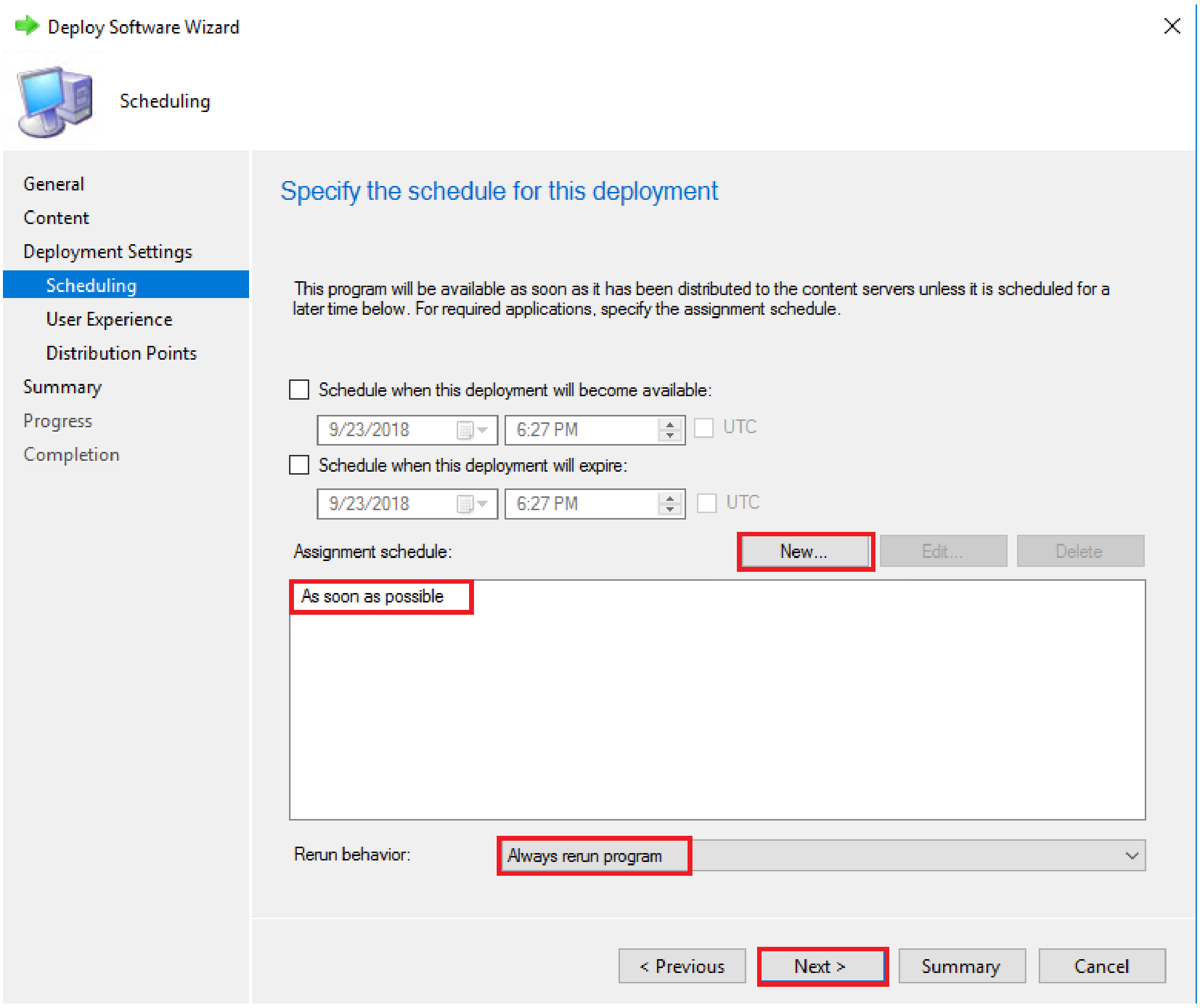
Purpose: Required

□ Pre-deploy software to the user's primary device

□ Send wake-up packets

□ Allow clients on a metered Internet connection to download content after the installation deadline, which might incur additional costs

5. Press "New" and assign "As soon as possible" as schedule, change rerun behavior to "Always rerun program". Click Next.



Specify the schedule for this deployment

This program will be available as soon as it has been distributed to the content servers unless it is scheduled for a later time below. For required applications, specify the assignment schedule.

□ Schedule when this deployment will become available:

9/23/2018 6:27 PM □ UTC

□ Schedule when this deployment will expire:

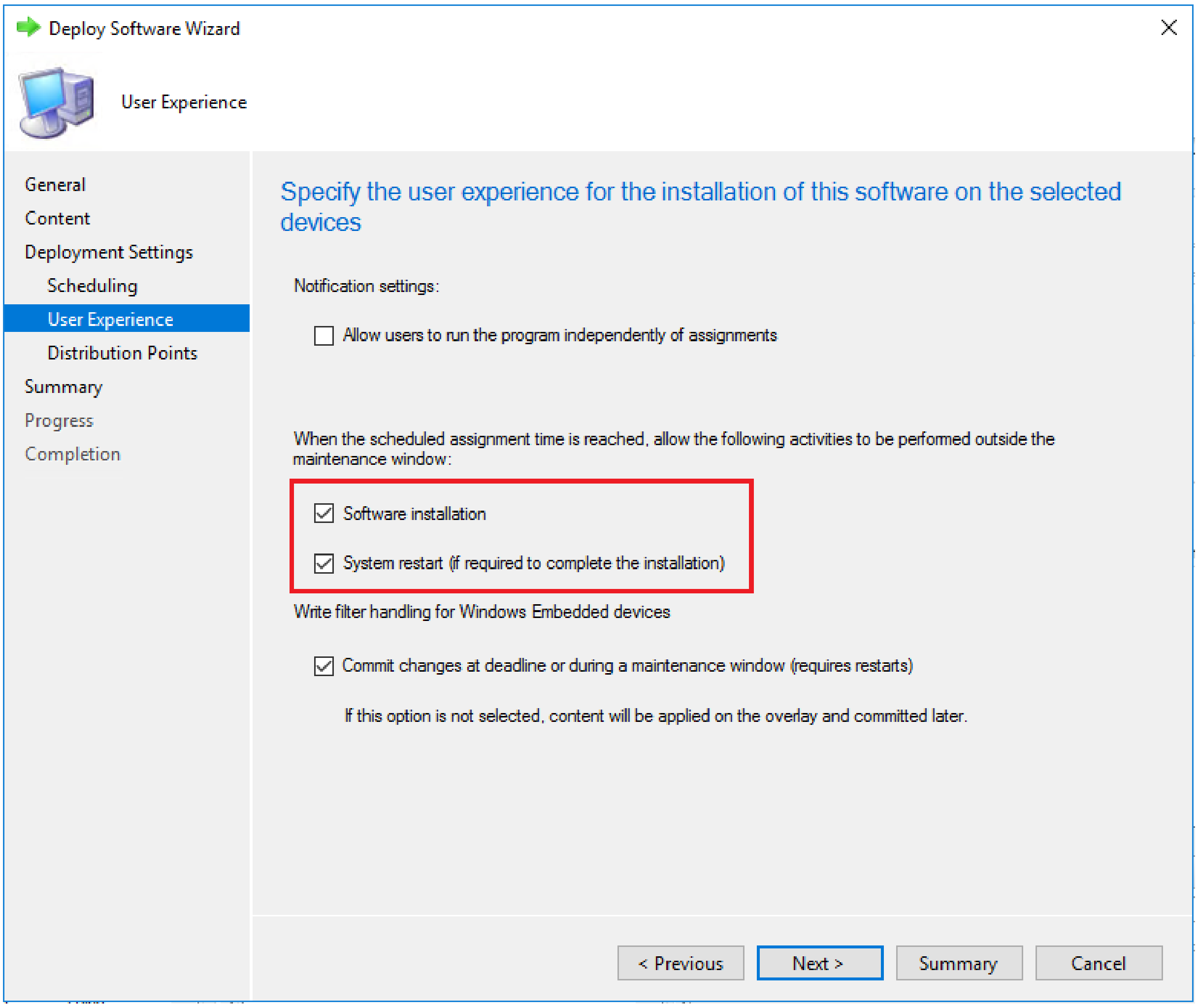
9/23/2018 6:27 PM □ UTC

Assignment schedule:

As soon as possible

Rerun behavior: Always rerun program

6. On the user experience tab, make sure the deployment can run outside maintenance windows by checking the two boxes in the middle. Click Next.



Specify the user experience for the installation of this software on the selected devices

Notification settings:

□ Allow users to run the program independently of assignments

When the scheduled assignment time is reached, allow the following activities to be performed outside the maintenance window:

☑ Software installation

☑ System restart (if required to complete the installation)

Write filter handling for Windows Embedded devices

☑ Commit changes at deadline or during a maintenance window (requires restarts)

If this option is not selected, content will be applied on the overlay and committed later.

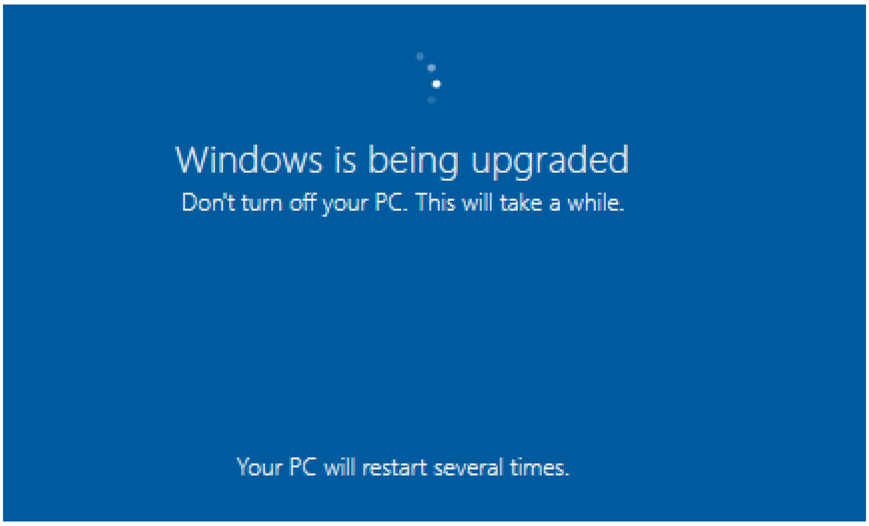
Remark: Checking these boxes is not crucial; but since TSLaunch, once launched the first time by the CM client, will allow the user to schedule the upgrade freely it's recommended for the simple reason of consistency.

7. We have no special recommendation for the "**Distribution Points**" settings for this deployment, configure according to your environmental situation. Click Next.
8. Click Next twice
9. Click Close.

11. UPGBACKGROUND

UPGBackground will cover the entire screen, regardless if a user is logged on or not.

In short this it is a full screen application that disables certain system key combinations such as ctrl-alt-del and Alt-F4. Except for a debug password, it requires no configuration.



Windows is being upgraded

Don't turn off your PC. This will take a while.

Your PC will restart several times.

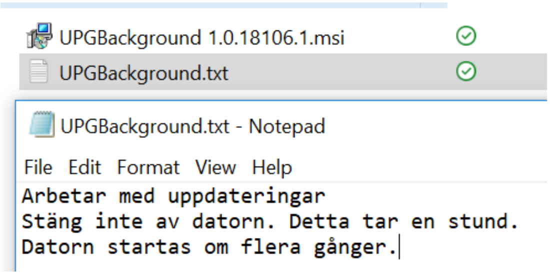
Remark: UPGBackground supports multiple monitors but will not detect a configuration change such as connecting an additional monitor (or open laptop lid) when already running.

11.1. Language support

UPGBackground shows only three lines of text. The language support is limited to one at the time.

Place a simple .txt file in the same folder as the msi and it will be copied along to the installations folder. UPGBackground will detect the file and try to use its content line for line.

Example (sv-SE):



UPGBackground 1.0.18106.1.msi

UPGBackground.txt

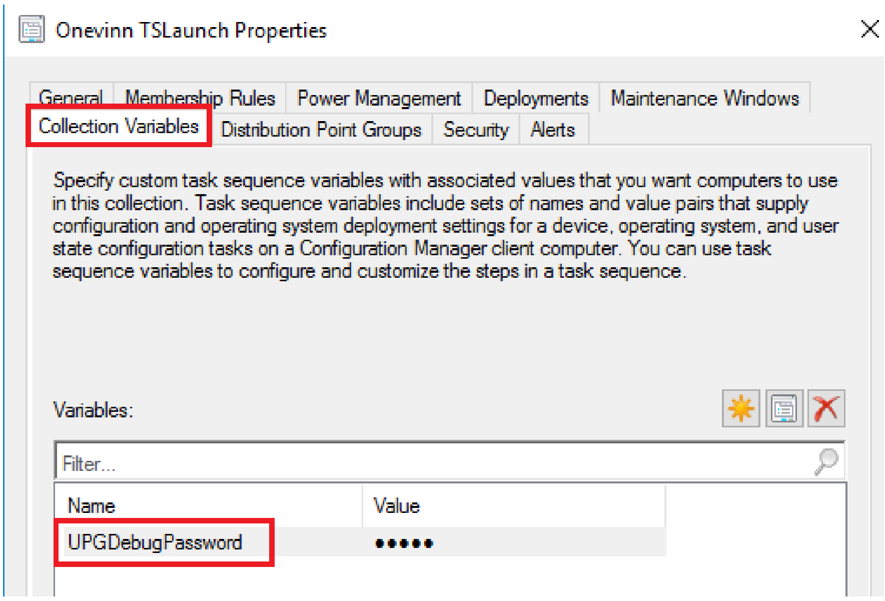
UPGBackground.txt - Notepad

```
Arbetar med uppdateringar
Stäng inte av datorn. Detta tar en stund.
Datorn startas om flera gånger.
```

Remark: The language file must be called exactly "**UPGBackground.txt**"

11.2. Debug mode

The debug password (exact spelling), **UPGDebugPassword**, is best set as a collection variable:



Onevinn TSLaunch Properties

Specify custom task sequence variables with associated values that you want computers to use in this collection. Task sequence variables include sets of names and value pairs that supply configuration and operating system deployment settings for a device, operating system, and user state configuration tasks on a Configuration Manager client computer. You can use task sequence variables to configure and customize the steps in a task sequence.

Variables:

Filter...

Name	Value
UPGDebugPassword	●●●●●

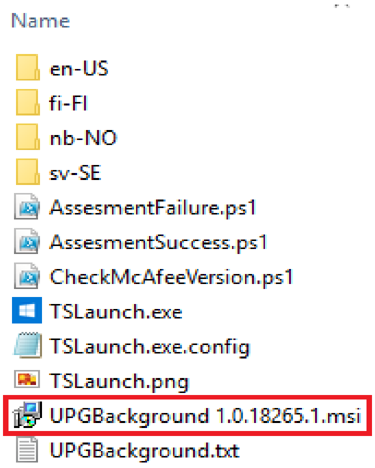
11.3. Install

“UPGBackground <version>.msi” can be installed in a “Install Application” step in the TS, the closer to the top the better.

However, if you are making use of the new “Pre-download content for this task sequence” functionality you will be even better off letting TSLaunch install it. Simply put the msi in the same folder as TSLaunch and set:

<add key="LaunchUPGBackground" value="**True**"/>

The later approach will launch UPGBackground significantly faster. In combination with a forced logoff it is very effective.



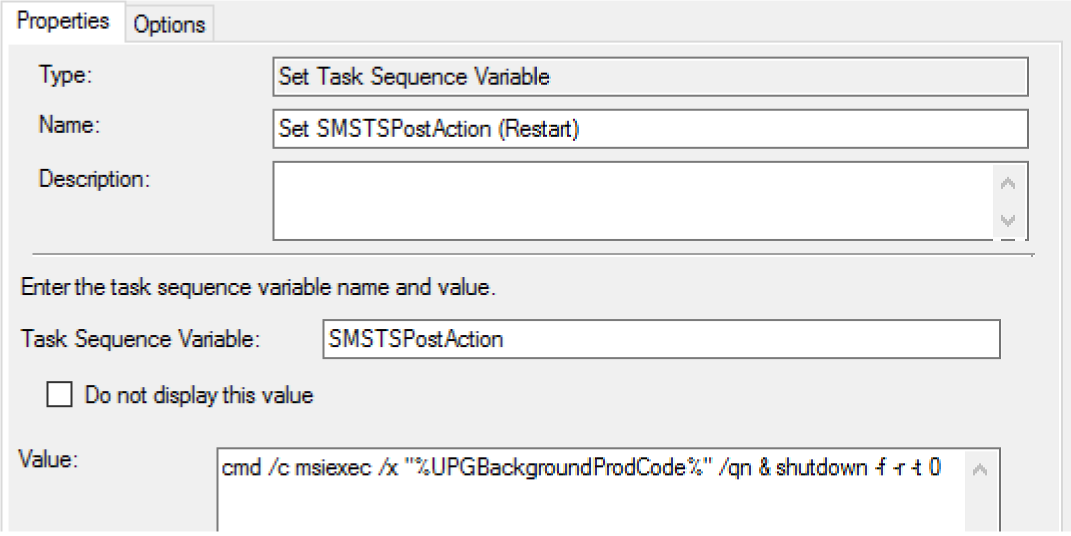
Name

- en-US
- fi-FI
- nb-NO
- sv-SE
- AssesmentFailure.ps1
- AssesmentSuccess.ps1
- CheckMcAfeeVersion.ps1
- TSLaunch.exe
- TSLaunch.exe.config
- TSLaunch.png
- UPGBackground 1.0.18265.1.msi
- UPGBackground.txt

11.4. Uninstall

Regardless of which method you pick for installation it will have to get uninstalled. The preferred method is to use **SMSTSPostAction**.

As soon as UPGBackground detects a running TS (TS-Environment) it will create a TS-variable named "**UPGBackgroundProdCode**" and set it to the msi's product code. We use the variable to set the SMSTSPostAction value.



Type: Set Task Sequence Variable

Name: Set SMSTSPostAction (Restart)

Description:

Enter the task sequence variable name and value.

Task Sequence Variable: SMSTSPostAction

□ Do not display this value

Value:

```
cmd /c msiexec /x "%UPGBackgroundProdCode%" /qn & shutdown -f -r -t 0
```

(In this example we have added a restart to run after the TS)

The step setting the variable SMSTSPostAction must be positioned **AFTER the last restart** in you task sequence. If you have implemented error handling (recommended) you will need to place a copy of the step also in the error group or figure out a logic that ensures that the step is always executed.

If it is not possible to use SMSTSPostAction a regular "Run Command Line" will do the job.

Command: cmd /c msiexec /x "%**UPGBackgroundProdCode%"** /qn

11.5. Self-destruct

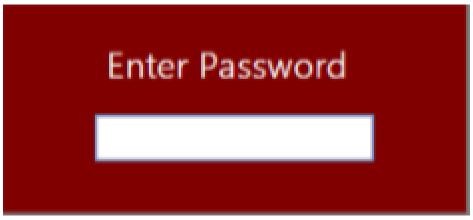
In the event the TS breaks, not fails, but completely breaks – the application has a built-in self-destruction function and will eventually (2-4 minutes) uninstall itself.

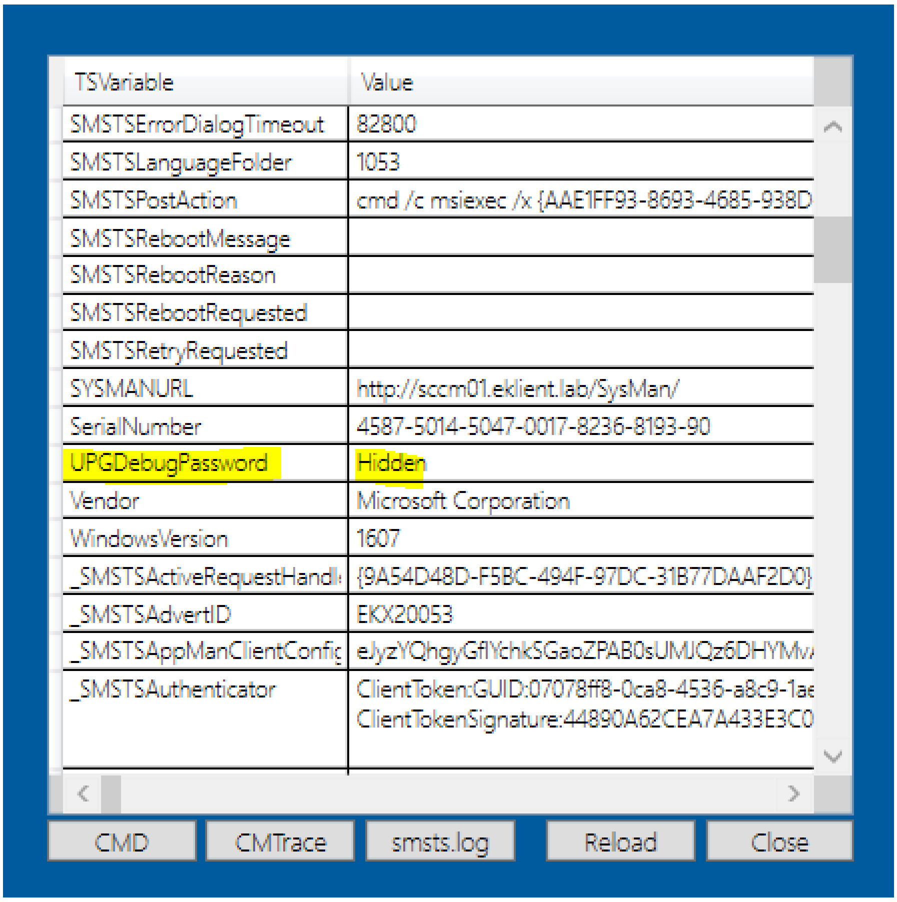
If a failure or misconfiguration causes the Task Sequence not to start at all UPGBackground will close and uninstall itself in 12-14 minutes.

11.6. Debug mode

Just as its older sibling “OSDBackground” UPGBackground can run in password protected debug mode. Right click the upper left corner and a password box will appear, provided that the TS has started, in other words, the password box might not appear just after a reboot.

Fill in the password stored in the Collection variable mentioned above and press enter.





TSVariable	Value
SMSTSErrorDialogTimeout	82800
SMSTSLanguageFolder	1053
SMSTSPostAction	cmd /c msiexec /x {AAE1FF93-8693-4685-938D
SMSTSRebootMessage	
SMSTSRebootReason	
SMSTSRebootRequested	
SMSTSRetryRequested	
SYSMANURL	http://sccm01.eklient.lab/SysMan/
SerialNumber	4587-5014-5047-0017-8236-8193-90
UPGDebugPassword	Hidden
Vendor	Microsoft Corporation
WindowsVersion	1607
_SMSTSActiveRequestHandl	{9A54D48D-F5BC-494F-97DC-31B77DAAF2D0}
_SMSTSAdvertID	EKX20053
_SMSTSAppManClientConfig	eJyzYQhgyGflYchkSGaoZPAB0sUMJQz6DHYMv
_SMSTSAuthenticator	ClientToken:GUID:07078ff8-0ca8-4536-a8c9-1ae ClientTokenSignature:44890A62CEA7A433E3C0

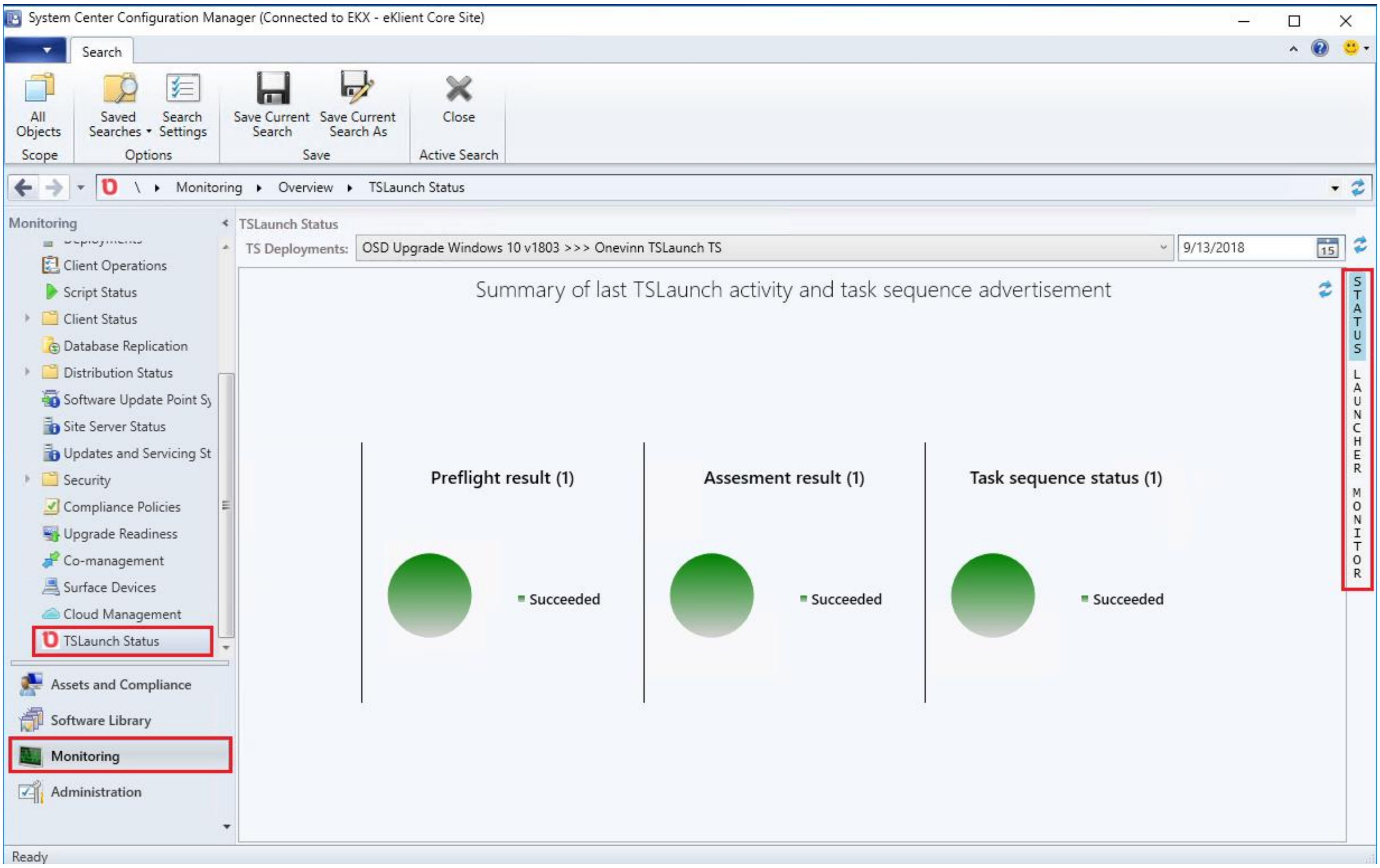
Known sensitive information is “**Hidden**”.

12. TSLAUNCH CONFIGMGR EXTENSION

TSLaunch will always log locally on the computer where it’s running. Additionally, one can configure it to write the same messages to a remote log file on a share, this is done in the config-file.

Apart from the above logging TSLaunch also sends custom status messages via the ConfigMgr client; these messages can be picked up from the site DB CM_XYZ or in WMI on the SMS provider.

To make the best use of this functionality it’s recommended to install the “TSLaunch ConfigMgr extension” which then appears as a new node under Monitoring in the ConfigMgr console.



TS Deployments: OSD Upgrade Windows 10 v1803 >>> Onevinn TSLaunch TS | 9/13/2018

Summary of last TSLaunch activity and task sequence advertisement

The extension has three main tabs on the right-hand side; STATUS, LAUNCHER and MONITOR.

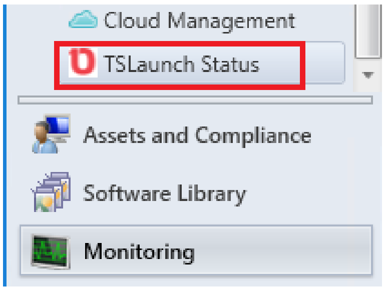
STATUS – shows three pie-chart for quick overview of your rollout status.

LAUNCHER – This tab has two sub tabs, one with last pre-flight results summary per client and a second with all results per client.

MONITOR – TS-monitoring. (This is not restricted to upgrade TS’s but can be used for any TS)

12.1. Installation

TSLaunch ConfigMgr Extension is installed as a single windows installer (msi). You can install it on any computer where the ConfigMgr console is previously installed. It will add a custom node under Monitoring. Simply double click the “TSLaunch ConfigMgr extension <version>.msi”, the version number is expected to change over time.

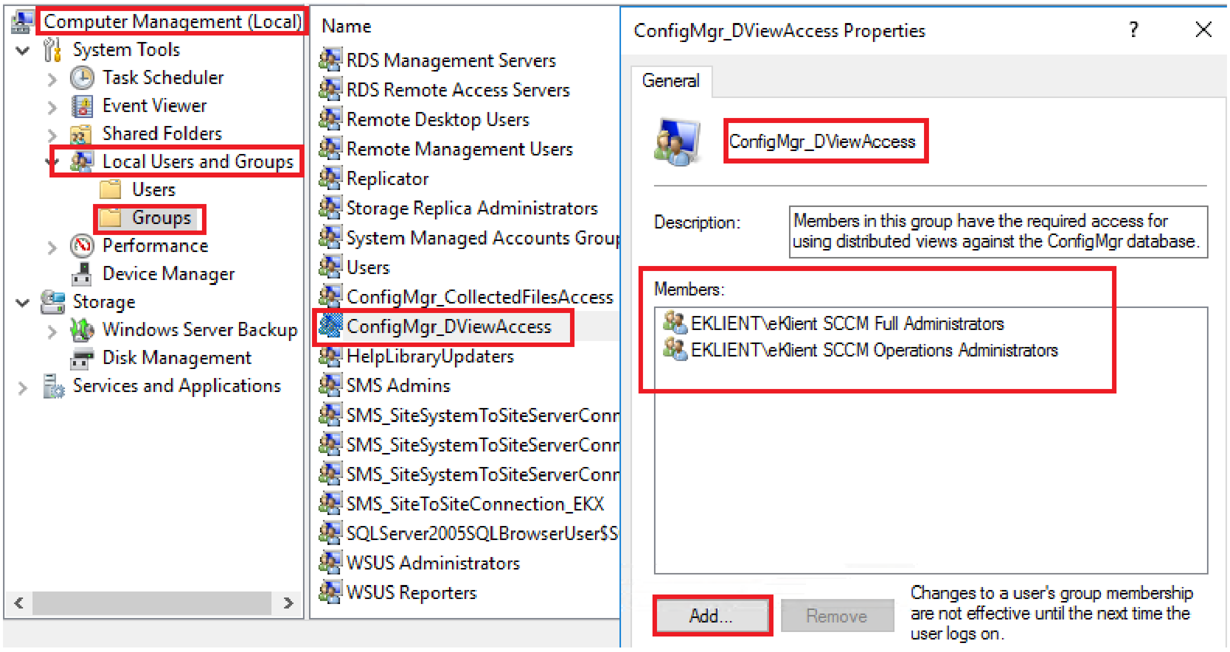


12.2. Permissions

The console extension queries the site database views; this is the only way to achieve good enough performance. A small issue is that the only ConfigMgr security role that is allowed access to the views by default is the Full Administrator. Furthermore, the user account cannot have been granted the role via group membership, the account must be explicitly added in the console.

To work around this problem all groups and or users that should be allowed to view the results presented by the extension must be added to a local security group on the server hosting the site database, typically the site server, in rare cases a remote SQL server.

Open the “Computer Management” console and add the appropriate users and AD groups to the local group “**ConfigMgr_DViewAccess**”.


Computer Management (Local)
System Tools
Task Scheduler
Event Viewer
Shared Folders
Local Users and Groups
Users
Groups
Performance
Device Manager
Storage
Windows Server Backup
Disk Management
Services and Applications
Name
RDS Management Servers
RDS Remote Access Servers
Remote Desktop Users
Remote Management Users
Replicator
Storage Replica Administrators
System Managed Accounts Grou
Users
ConfigMgr_CollectedFilesAccess
ConfigMgr_DViewAccess
HelpLibraryUpdaters
SMS Admins
SMS_SiteSystemToSiteServerConn
SMS_SiteSystemToSiteServerConn
SMS_SiteSystemToSiteServerConn
SMS_SiteToSiteConnection_EKX
SQLServer2005SQLBrowserUser$S
WSUS Administrators
WSUS Reporters
ConfigMgr_DViewAccess Properties
?
General
Add...
Remove

ConfigMgr_DViewAccess

Description: Members in this group have the required access for using distributed views against the ConfigMgr database.

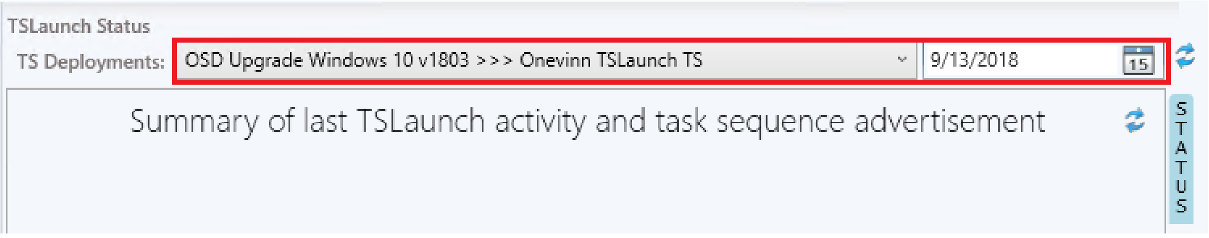
Members:
EKLIENT\eKlient SCCM Full Administrators
EKLIENT\eKlient SCCM Operations Administrators

Changes to a user's group membership are not effective until the next time the user logs on.

12.3. Choosing deployment

Before you can see any actual statistics or status you must choose a deployment. This is done from the drop-down combo box at the top.

In this case we are looking at the result from a task sequence named “**OSD Upgrade Windows 10 v1803**” which has been **deployed to** a collection named “**Onevinn TSLaunch TS**”. Per default the result is limited to the last 10 days, you can change that in the date picker to the right.



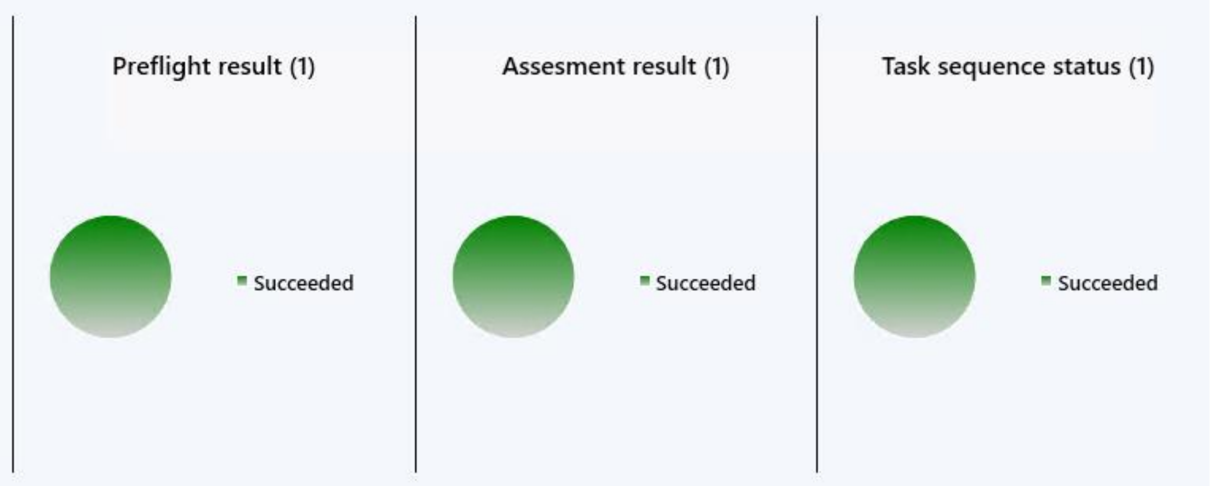
TSLaunch Status

TS Deployments: OSD Upgrade Windows 10 v1803 >>> Onevinn TSLaunch TS | 9/13/2018

Summary of last TSLaunch activity and task sequence advertisement

12.4. Status pies

On the STATUS tab you will find a visual overview of the status of you upgrade rollout. It will be limited to the timespan indicated in the date picker, and of course based on the deployment chosen in the drop-down box only.

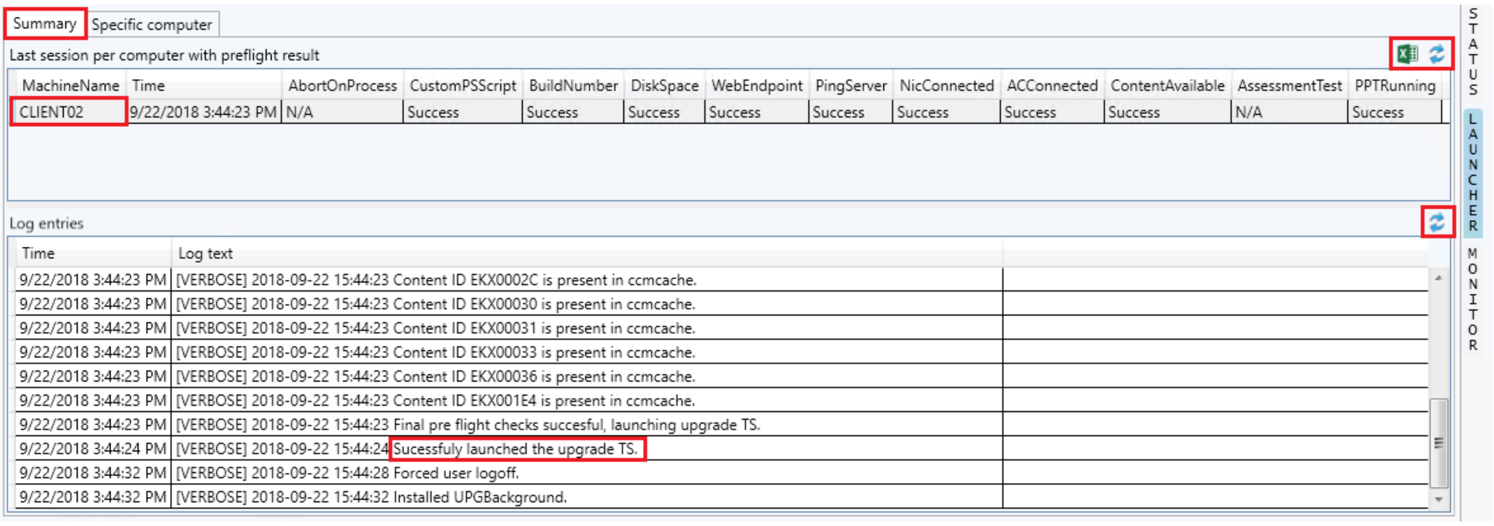


We leave the interpretation of these results to you; it’s likely that it will look very different depending on how TSLaunch has been configured, if assessment tests are not run, as an example, only two pie charts will be shown.

12.5. Launcher

On the “**Summary**” tab you will see the result of the **last preflight check** per computer in the deployment. When a row is clicked in the upper grid the complete log for that session will appear in the lower grid.

The grids can be refreshed separately and the upper exported to a csv file.



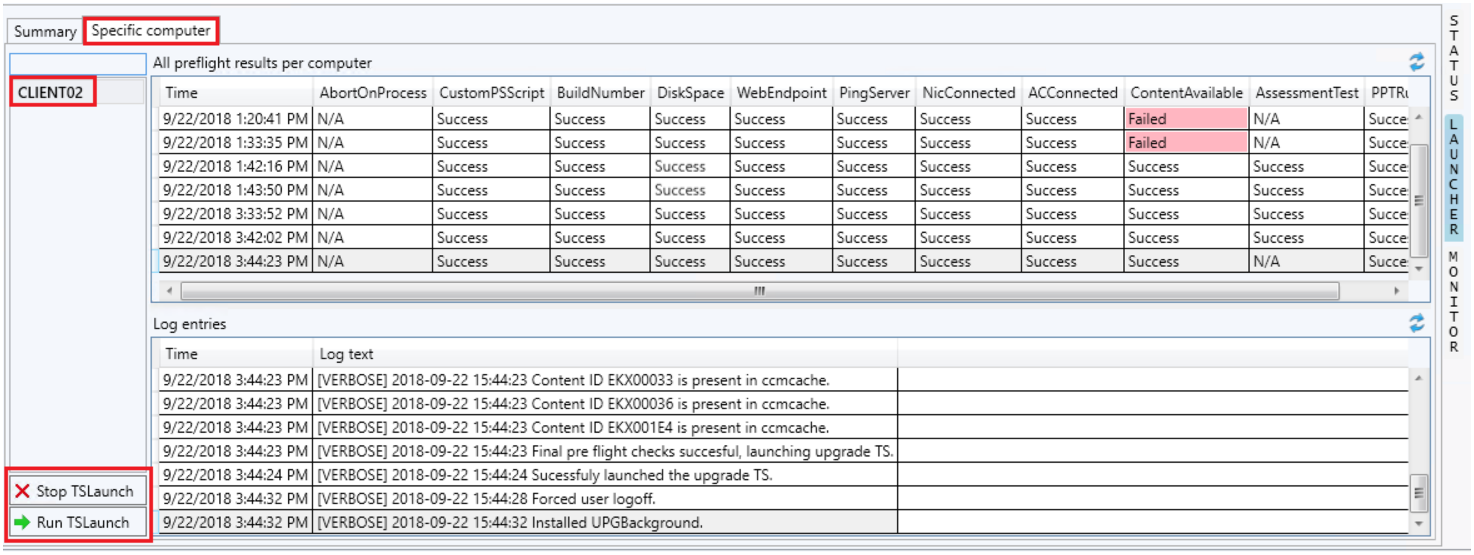
Last session per computer with preflight result

MachineName	Time	AbortOnProcess	CustomPSScript	BuildNumber	DiskSpace	WebEndpoint	PingServer	NicConnected	ACConnected	ContentAvailable	AssessmentTest	PPTRunning
CLIENT02	9/22/2018 3:44:23 PM	N/A	Success	Success	Success	Success	Success	Success	Success	Success	N/A	Success

Log entries

Time	Log text
9/22/2018 3:44:23 PM	[VERBOSE] 2018-09-22 15:44:23 Content ID EKX0002C is present in ccmcache.
9/22/2018 3:44:23 PM	[VERBOSE] 2018-09-22 15:44:23 Content ID EKX00030 is present in ccmcache.
9/22/2018 3:44:23 PM	[VERBOSE] 2018-09-22 15:44:23 Content ID EKX00031 is present in ccmcache.
9/22/2018 3:44:23 PM	[VERBOSE] 2018-09-22 15:44:23 Content ID EKX00033 is present in ccmcache.
9/22/2018 3:44:23 PM	[VERBOSE] 2018-09-22 15:44:23 Content ID EKX00036 is present in ccmcache.
9/22/2018 3:44:23 PM	[VERBOSE] 2018-09-22 15:44:23 Content ID EKX001E4 is present in ccmcache.
9/22/2018 3:44:23 PM	[VERBOSE] 2018-09-22 15:44:23 Final pre flight checks succesful, launching upgrade TS.
9/22/2018 3:44:24 PM	[VERBOSE] 2018-09-22 15:44:24 Sucessfuly launched the upgrade TS.
9/22/2018 3:44:32 PM	[VERBOSE] 2018-09-22 15:44:28 Forced user logoff.
9/22/2018 3:44:32 PM	[VERBOSE] 2018-09-22 15:44:32 Installed UPGBackground.

On the “**Specific computer**” tab you can drill further down and analyze the result of not only the last, but all preflight checks ran on a specific computer.



All preflight results per computer

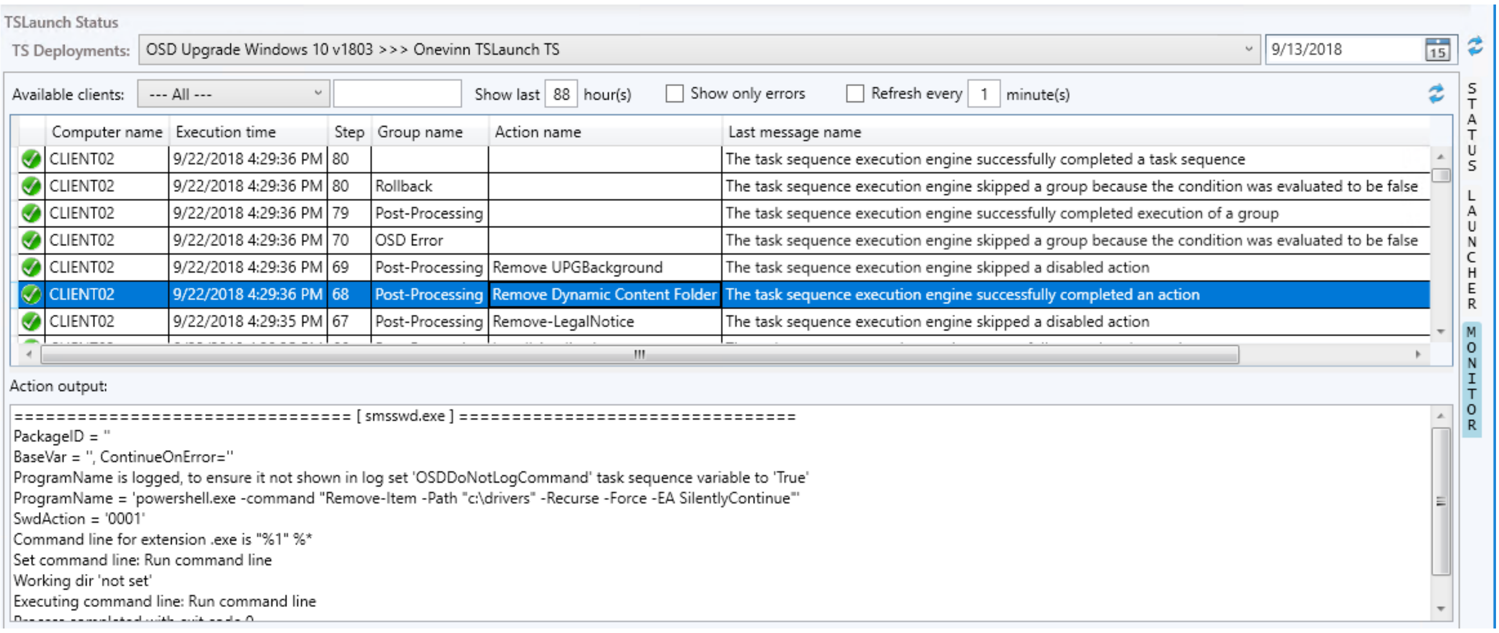
Time	AbortOnProcess	CustomPSScript	BuildNumber	DiskSpace	WebEndpoint	PingServer	NicConnected	ACConnected	ContentAvailable	AssessmentTest	PPTRı
9/22/2018 1:20:41 PM	N/A	Success	Success	Success	Success	Success	Success	Success	Failed	N/A	Succe
9/22/2018 1:33:35 PM	N/A	Success	Success	Success	Success	Success	Success	Success	Failed	N/A	Succe
9/22/2018 1:42:16 PM	N/A	Success	Success	Success	Success	Success	Success	Success	Success	Success	Succe
9/22/2018 1:43:50 PM	N/A	Success	Success	Success	Success	Success	Success	Success	Success	Success	Succe
9/22/2018 3:33:52 PM	N/A	Success	Success	Success	Success	Success	Success	Success	Success	Success	Succe
9/22/2018 3:42:02 PM	N/A	Success	Success	Success	Success	Success	Success	Success	Success	Success	Succe
9/22/2018 3:44:23 PM	N/A	Success	Success	Success	Success	Success	Success	Success	Success	N/A	Succe

Log entries

Time	Log text
9/22/2018 3:44:23 PM	[VERBOSE] 2018-09-22 15:44:23 Content ID EKX00033 is present in ccmcache.
9/22/2018 3:44:23 PM	[VERBOSE] 2018-09-22 15:44:23 Content ID EKX00036 is present in ccmcache.
9/22/2018 3:44:23 PM	[VERBOSE] 2018-09-22 15:44:23 Content ID EKX001E4 is present in ccmcache.
9/22/2018 3:44:23 PM	[VERBOSE] 2018-09-22 15:44:23 Final pre flight checks succesful, launching upgrade TS.
9/22/2018 3:44:24 PM	[VERBOSE] 2018-09-22 15:44:24 Sucessfuly launched the upgrade TS.
9/22/2018 3:44:32 PM	[VERBOSE] 2018-09-22 15:44:28 Forced user logoff.
9/22/2018 3:44:32 PM	[VERBOSE] 2018-09-22 15:44:32 Installed UPGBackground.

In the lower left corner there are two buttons allowing you to either completely dismiss TSLaunch on the specific computer or rerun it immediately.

Finally, on the “**MONITOR**” tab you will be able to monitor your task sequences. This is a shameless full rip-off from Trevor Jones magnificent “ConfigMgr Task Sequence Monitor” tool also available on Technet Gallery (https://gallery.technet.microsoft.com/ConfigMgr-Task-Sequence-fefdc532). We recommend it for stand-alone scenarios or if you suffer from the need to monitor MDT Task sequences, which the Onevinn extension does not, and will not, support.



TSLaunch Status

TS Deployments: OSD Upgrade Windows 10 v1803 >>> Onevinn TSLaunch TS 9/13/2018

Available clients: --- All --- Show last 88 hour(s) □ Show only errors □ Refresh every 1 minute(s)

	Computer name	Execution time	Step	Group name	Action name	Last message name
✓	CLIENT02	9/22/2018 4:29:36 PM	80			The task sequence execution engine successfully completed a task sequence
✓	CLIENT02	9/22/2018 4:29:36 PM	80	Rollback		The task sequence execution engine skipped a group because the condition was evaluated to be false
✓	CLIENT02	9/22/2018 4:29:36 PM	79	Post-Processing		The task sequence execution engine successfully completed execution of a group
✓	CLIENT02	9/22/2018 4:29:36 PM	70	OSD Error		The task sequence execution engine skipped a group because the condition was evaluated to be false
✓	CLIENT02	9/22/2018 4:29:36 PM	69	Post-Processing	Remove UPGBackground	The task sequence execution engine skipped a disabled action
✓	CLIENT02	9/22/2018 4:29:36 PM	68	Post-Processing	Remove Dynamic Content Folder	The task sequence execution engine successfully completed an action
✓	CLIENT02	9/22/2018 4:29:35 PM	67	Post-Processing	Remove-LegalNotice	The task sequence execution engine skipped a disabled action

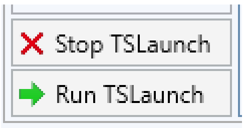
Action output:

```
================================ [ smsswd.exe ] ================================
PackageID = ''
BaseVar = '', ContinueOnError=''
ProgramName is logged, to ensure it not shown in log set 'OSDDoNotLogCommand' task sequence variable to 'True'
ProgramName = 'powershell.exe -command "Remove-Item -Path "c:\drivers" -Recurse -Force -EA SilentlyContinue"'
SwdAction = '0001'
Command line for extension .exe is "%1" %*
Set command line: Run command line
Working dir 'not set'
Executing command line: Run command line
Process completed with exit code 0
```

13. CONSOLE SCRIPTS

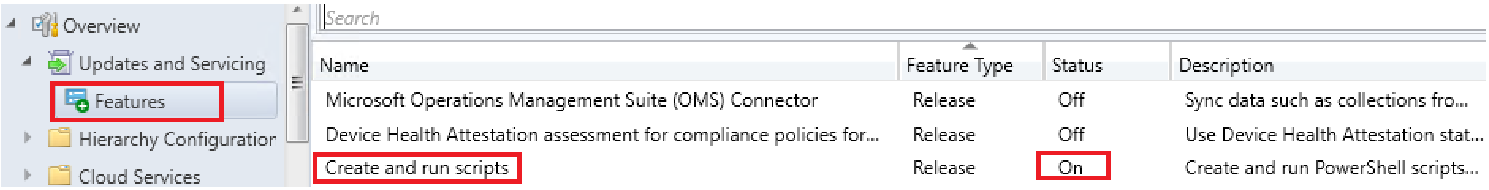
The two magic buttons on the LAUNCHER “Specific computer” tab deserves a separate explanation. When any of these buttons is pressed for the first time a corresponding script will be created on the script’s node under “Software Library > Overview > Scripts” and run against the chosen computer.

Buttons



Required feature

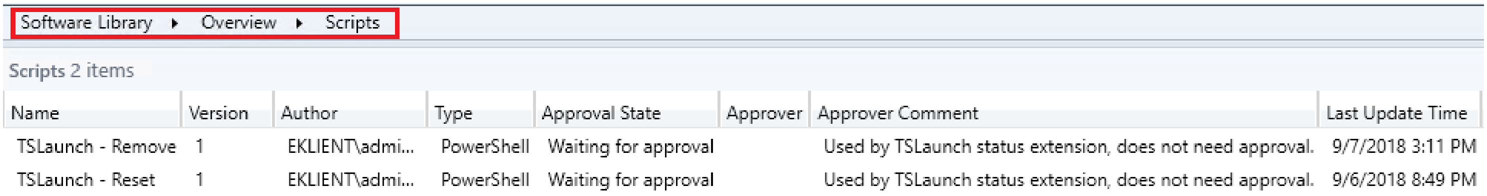
If the “Create and run scripts” feature isn’t available a message box will inform about it. You can then turn it on here (available as preview since a couple of CB’s, released with 1806).



Name	Feature Type	Status	Description
Microsoft Operations Management Suite (OMS) Connector	Release	Off	Sync data such as collections fro...
Device Health Attestation assessment for compliance policies for...	Release	Off	Use Device Health Attestation stat...
Create and run scripts	Release	On	Create and run PowerShell scripts...

Scripts

In case the feature is turned on the scripts are created. They do not need approval, as they can be run from code regardless of being approved or not.



Scripts 2 items

Name	Version	Author	Type	Approval State	Approver	Approver Comment	Last Update Time
TSLaunch - Remove	1	EKLIENT\admi...	PowerShell	Waiting for approval		Used by TSLaunch status extension, does not need approval.	9/7/2018 3:11 PM
TSLaunch - Reset	1	EKLIENT\admi...	PowerShell	Waiting for approval		Used by TSLaunch status extension, does not need approval.	9/6/2018 8:49 PM

In the event you decide not to use the console extension and uninstalls it, the scripts remain and will have to be removed (deleted) manually.

14. HOW TSLAUNCH WORKS - OVERVIEW

When TSLaunch is first run, by the CM Client, it immediately creates a schedule task. This task is configured to relaunch the application once a day, either 15 minutes after logon or at 10h00, which ever happens first. The purpose of this task is to guarantee that the process doesn't fail, even if other problems occurs or the user finds a cunning way of avoiding scheduling the upgrade.

TSLaunch is run only once by the CM Client.

The idea is to establish contact with the user as soon as possible, when all prereqs (like content available on disk) are positive TSLaunch will create a desktop enabled process and ask the user to either upgrade immediately or schedule the upgrade. This is done by a second schedule task that reruns the application twice an hour until success.

When a user chooses to schedule the upgrade the "twice an hour task" is dismissed, instead a task is created that launches the application according to what the user requested. What happens when this custom task is triggered is depending on whether the user choose to upgrade directly or requested to just be notified.

TSLaunch stores some information in the registry under HKLM\Software\Onevinn\TSLaunch.

When the Task Sequence is finally launched all scheduled tasks and registry values are removed.

15. ISSUES AND SUCH

Requirements:

DotNet 4.6.2 or higher.

The console extension “only” requires DotNet 4.5.2, to be consistent with the console it-self.

Issues:

You might want to configure your TSLaunch package as well as TS content and media to be persisted in the CCMCache, to avoid problems due to it being cycled out.

Note that status messages are not persistent, ConfigMgr will “clean” them out after aprox. 30 days. This means the console extension gives a good overview of the present status of your upgrades. If you need history to be stored for any length of time, make sure to configure remote logging to a file share**. See 9.4 LogLocation for details**.

16. CHANGELOG

2018-10-28 - TSLaunch 2.3.18300.1:

- Bugfix: PowerPoint slide-show not detected.
- Bugfix: Exception when opening scheduler on "EndDate".
- Added logging of UI restore from system tray and remaining days to upgrade.

2018-10-28 - TSLaunch ConfigMgr extension 2.3.18301.1:

- Bugfix: Computer list on "Specific computer" tab not updated on refresh.
- All grids are now sorted in descending order on execution time.
- Added possibility to clear status history for single computer and/or remove specific status entry (Right click).
- Added console scripts to uninstall UPGBackground remotely.
- Task sequence monitor is now considering exit code "3010" as success.

2018-10-28 - UPGBackground 1.0.18301.1:

- Stores the Debug password (encrypted) to registry making debug mode available also when TSManager is not running.

2018-10-14 - TSLaunch ConfigMgr extension 2.3.18317.1:

- Bugfix: Script feature not detected correctly on non en-US systems.

2018-11-19 – TSLaunch 2.3.18321.1

- Added new config file key: making it possible to run a script when the TS is launched.

2018-11-19 – UPGBackground 1.0.18321.1

- Added multi language support, add a prefix to the file name like “sv-SE_” or “fi-FI_”

2021-01-18 - TSLaunch 2.4.21014.01

- Added possibility to reduce the number of status messages sent to ConfigMgr
- Added AC connected Dialog.
- Added Toast reminder twice on the of scheduled upgrade.
- Bugfix: If the user logged out whilst the application was minimized it sometimes didn’t run again.