

# Contents

## Customize

### Desktop customizations

Taskbar

Start layout

### Out of Box Experience (OOBE)

OOBE.xml

Cortana voice support

OOBE screen details

Windows updates during OOBE

OEM HID pairing

OEM license terms

OEM registration pages

    Design your registration pages

    Configure OOBE.xml

    Protect and collect user data

Automate OOBE

Retail demo experience

Windows performance power slider

Dark mode

Get Help app

SIM card slot names

Mobile broadband: SAR table

Pen and Windows ink

Enterprise desktop customizations

    WEDL\_AssignedAccess

    Custom Logon

        Complementary features to Custom Logon

        Troubleshooting Custom Logon

    Keyboard Filter

[Keyboard Filter key names](#)

[Predefined key combinations](#)

[Keyboard Filter WMI provider reference](#)

[WEKF\\_CustomKey](#)

[WEKF\\_PredefinedKey](#)

[WEKF\\_Scancode](#)

[WEKF\\_Settings](#)

[Windows PowerShell script samples for Keyboard Filter](#)

[Add blocked key combinations](#)

[Disable all blocked key combinations](#)

[List all configured key combinations](#)

[Modify global settings](#)

[Remove key combination configurations](#)

[Shell Launcher](#)

[WESL\\_UserSetting](#)

[WESL\\_UserSetting.GetCustomShell](#)

[WESL\\_UserSetting.RemoveCustomShell](#)

[WESL\\_UserSetting.SetCustomShell](#)

[WESL\\_UserSetting.SetDefaultShell](#)

[WESL\\_UserSetting.GetDefaultShell](#)

[WESL\\_UserSetting.IsEnabled](#)

[WESL\\_UserSetting.SetEnabled](#)

[Unbranded Boot](#)

[Unified Write Filter \(UWF\) feature](#)

[Hibernate Once/Resume Many \(HORM\)](#)

[Write filter exclusions](#)

[Overlay location and size](#)

[Turn on UWF](#)

[Service UWF-protected devices](#)

[Antimalware support on UWF-protected devices](#)

[Apply Windows updates to UWF-protected devices](#)

[Apply OEM updates to UWF-protected devices](#)

[UWF master servicing script](#)  
[UWF servicing screen saver](#)  
[Troubleshooting Unified Write Filter \(UWF\)](#)  
[Unified Write Filter WMI provider reference](#)

[UWF\\_ExcludedFile](#)  
[UWF\\_ExcludedRegistryKey](#)  
[UWF\\_Filter](#)  
[UWF\\_Overlay](#)  
[UWF\\_RegistryFilter](#)  
[UWF\\_Servicing](#)  
[UWF\\_Volume](#)  
[uwfmgr.exe](#)

[Windows System Image Manager Technical Reference](#)

[Overview](#)

[Scenarios Overview](#)  
[User Interface Overview](#)  
[Windows Image Files and Catalog Files Overview](#)  
[Answer Files Overview](#)  
[Best Practices for Authoring Answer Files](#)  
[Distribution Shares and Configuration Sets Overview](#)

[How-to Topics](#)

[Open a Windows Image or Catalog File](#)  
[Create or Open an Answer File](#)  
[Configure Components and Settings in an Answer File](#)  
[Validate an Answer File](#)  
[Hide Sensitive Data in an Answer File](#)  
[Add a Device Driver Path to an Answer File](#)  
[Add a Package to an Answer File](#)  
[Add a Custom Command to an Answer File](#)  
[Find a Component, Setting, or Package in Windows SIM](#)  
[Create a Configuration Set](#)  
[Create or Open a Distribution Share](#)

[Manage Files and Folders in a Distribution Share](#)

[Add Packages to a Distribution Share](#)

[Reference](#)

[Component Settings and Properties Reference](#)

[Windows System Image Manager Architecture](#)

[Windows System Image Manager Supported Platforms](#)

[Unattended Windows Setup Reference](#)

[Mobile customizations](#)

[Enterprise shared storage](#)

[Customize using the mobile MCSF framework](#)

[Managed Centralized Settings Framework \(MCSF\)](#)

[Customization answer file](#)

[Set phone metadata in DeviceTargetingInfo](#)

[Set languages and locales](#)

[Create a resource-only .dll for localized strings](#)

[Customizations for device management](#)

[Enabling runtime configuration](#)

[Managing runtime configuration data](#)

[Override the default CountryTable.xml](#)

[Setting the UICC slot for branding configuration](#)

[Customizations for hardware components](#)

[Buttons: Enabling the Start button to wake the phone](#)

[Camera: Improved user experience for phones without a HW camera button](#)

[Display: Building images for FWVGA panels with static software buttons](#)

[Display: Building images with user-managed software buttons](#)

[Networking: Configuring the MTU data size](#)

[Sensors: Auto brightness](#)

[Storage: Enabling the packed commands feature for eMMC](#)

[Storage: Enabling the UHS-1 feature for SD cards](#)

[Storage: Enabling the HS200 feature for eMMC](#)

[Touch: Defining capacitive button behavior](#)

[Touch: Describing the physical width and height of the display](#)

[Touch: Specifying the repeat rate for touch samples during touch-and-hold presses](#)

[Customizations for applications and Microsoft components](#)

[Active phone cover settings](#)

[Customize the SIM toolkit](#)

[Enhanced apps experience for medium and large screens](#)

[Include required Microsoft components to the image](#)

[Phone call/SMS filter applications](#)

[Preload an app with a dependency](#)

[Remove optional Microsoft components from the image](#)

[Store live tile](#)

[Customizations for boot, initial setup, and shutdown](#)

[Configure the timezone confirmation page during setup](#)

[Configuring a boot screen to display in the final boot screen slot](#)

[Configuring boot battery charging behavior](#)

[Configuring OEM and mobile operator boot screens](#)

[Configuring the duration of the first boot screen](#)

[Custom shutdown screen](#)

[Language selection during initial setup](#)

[Partner account configuration during setup](#)

[Screen background color during initial setup](#)

[Set the default country/region when SIM PIN is on](#)

[Customizations for browser](#)

[Custom HTTP headers for Microsoft Edge](#)

[Custom user agent string for Microsoft Edge](#)

[Default value for browser data saver](#)

[Show pictures automatically when browsing](#)

[Welcome home page for Microsoft Edge](#)

[WinInet ReceiveTimeOut duration](#)

[Customizations for Cellular connectivity](#)

[Background cellular data restriction](#)

[Cellular data connection icon](#)

[Connection speed option](#)

- Custom percentages for signal strength bars
  - Data transfer indicator
  - Default highest connection speed
  - Default roaming option
  - Disable Cell Broadcast
  - Extended error messages for reject codes
  - Hide CDMA mode selection
  - Hide Cellular & SIM Settings
  - LTE attach: GUID for user configured internet APN
  - LTE attach: Mapping OEMConnectionId values to modem profiles
  - Manual network selection timeout
  - Maximum number of PDP contexts
  - Permanent automatic mode
  - Preferred data provider list
  - Remove cellular functionality from the device
  - Roaming filter
- Customizations for Wi-Fi settings and connectivity
- Authentication for Wi-Fi hotspot settings
  - Cellular data fallback when in limited Wi-Fi connectivity
  - Change Wi-Fi to WLAN
  - Connecting to open Wi-Fi hotspots in Windows 10
  - Enable static IP
  - Limited connectivity status
  - Wi-Fi always on, always connected
  - Wi-Fi calling errors
  - Wi-Fi calling operator name
  - Wi-Fi icon and notifications
- Customizations for contacts
- Cortana phone number
  - Disable wait for phonebook ready signal from the modem
  - Hide contacts without phone numbers
  - Sort order for contacts

Sort order for contacts override

Customizations for desktop experiences

Control Panel device icon

Phone image in the phone app

Customizations for display and lock screen

Additional lock screen backgrounds

Brightness tuning

Default theme and accent color for Kid's Corner

Enable dark mode

Hide the auto brightness setting

Lock screen notifications

Lock screen timeout for AMOLED and OLED displays

Warning about light theme for AMOLED and OLED screens

Customizations for email

Light or dark theme in email

Customizations for keyboard

Disable text correction and suggestions

Hardware keyboard character repeats hold time and delay

On-screen keyboard delay

Pre-enabled keyboard

Text correction and suggestions

Customizations for maps

Map data on an SD card and map preload

Maps for phones shipped in China

Preloaded map data in the user store

Temporary map data cache size

Customizations for notifications and quick actions

Add an LED notification option

Configure Quick actions

CMAS Required Monthly Test

Display CMAS message order

Emergency notifications

## Customizations for phone calls

- Adjust the call duration information for CDMA calls
- Always display the dialed phone number
- Branding for phone calls
- Caller ID matching
- Cause codes
- Conditional call forwarding
- Configure DTMF tones
- Configure message waiting indicator notifications
- Dialer codes for supplementary services
- Dialer codes to launch diagnostic applications
- Dial string overrides when roaming
- Disable link to contact card in active call screen
- Disable video upgrade Store navigation
- Disable voicemail phone number display
- Dismiss the last USSD waiting dialog
- Emergency phone numbers
- Enable call recording by default
- Enable IMS services
- Enable RCS
- Hide call forwarding
- Maximum number of participants in a VoLTE conference call
- Network-controlled caller ID settings
- Override the voicemail number on the UICC
- Supplementary services exclusions
- Trim supplementary service codes
- Use OK for USSD dialogs
- Use HD audio codec for call branding
- Use voice domain for emergency call branding
- Visual voicemail
- Voice over LTE call indication
- Voicemail number for CDMA phones

## Customizations for photos, music, and videos

Adding OEM lens apps as options for the default camera

Audio volume limitation

Configure OEM lens apps to launch above the lock screen

Configure the FM radio

Maximum enumerable photo size

Reset the audio volume limitation and warning

Settings for capture mode, burst support, and burst storage duration

Video over LTE

## Customizations for ringtones + sounds

Additional alarms

Additional notification sounds

Additional ringtones

Call drop and call waiting sounds

Camera shutter sound

Ringtone store application

## Customizations for SMS and MMS

Add encoding extension tables for SMS

Automatic send retry and resize settings for MMS messages

Automatically retry downloading MMS messages

Content location in the multimedia message service center (MMSC)

Delay for resend attempts

Disable editing of the SMS center number

Disable the EMS long messages feature

Expiration time for SMS messages

Extract phone numbers in strings

Full error messages for SMS and MMS

IMS retry

IMSI authentication

International assisted dialing for SMS

Maximum length for SMS messages

Maximum number of attachments for MMS messages

- Maximum number of recipients for SMS and MMS
- MMS APN settings
- MMS automatic download
- MMS data options
- MMS delivery confirmation
- MMS for group messages
- MMS receipt acknowledgement
- Permanent SMS message failures
- Ports that accept cellular broadcast messages
- Proxy authorization for MMS
- Select multiple recipients for SMS and MMS messages
- Send SMS messages to SMTP addresses
- Server for MMS acknowledgement messages
- SMS delivery confirmation
- SMS encoding
- SMS intercept deny list
- SMS intercept ports
- Support HTTP cache-control no-transform for MMS
- Supported protocols for service indication messages
- Switch from SMS to MMS for long messages
- Truncated content handling for WAP push notification
- Use insert-address-token or local raw address
- Use UTF-8 for MMS messages with unspecified character encoding
- User agent profile for MMS messages
- User agent string for MMS messages
- User alert for service indication messages
- Video attachments in MMS
- Voicemail SMS intercept
- Customizations for SIM settings
  - Add a suffix to the mobile operator name
  - Additional Internet APN settings
  - Change SIM to SIM/UIM

Change the default SIM name to match the SPN or operator name

Configure C+G dual SIM settings

Hide the SIM security settings option

Remove the trailing MSISDN digits on a SIM card

Settings for IMS services

View Internet APN

Customizations for locale-based settings

Assistance for dialing international phone numbers

China Type Approval requirement: app install prompts

Contact management on the SIM (CN only)

Disable NITZ or daylight saving time

Display location icon

Ignore NITZ information from LTE networks

Microsoft Store for China

Mobile device languages

Network Time Protocol support

Regional format

Speech languages

Default list of countries/regions

Preferred system types for phone connectivity (CN only)

Threshold for automatic time update

Time zone priority list

WAP browser support (CN and IN only)

Customizations for accessibility settings

Telecoil and TTY support for accessibility

Customizations for phone update settings

Auto scan for phone updates

Block using SD card for updates

Enable SD card override

Customizations for USB settings

Enable the incompatible charger notification

Enable the data connection prompt

[Hide the weak charger notification option UI](#)

[Registry values for mobile operator IDs](#)

[Registry values for carrier-unlocked phones](#)

[Power settings](#)

[Adaptive hibernate](#)

[StandbyBudgetPercent](#)

[StandbyReserveTime](#)

[Power controls](#)

[EnableInputSuppression](#)

[IgnoreCsComplianceCheck](#)

[LidNotificationsAreReliable](#)

[Processor power management options](#)

[Static configuration options for core parking](#)

[CPMinCores](#)

[CPMaxCores](#)

[CPIIncreaseTime](#)

[CPDecreaseTime](#)

[CPConcurrency](#)

[CPDistribution](#)

[CPHeadroom](#)

[CpLatencyHintUnpark](#)

[Static configuration options for the performance state engine](#)

[MaxPerformance](#)

[MinPerformance](#)

[PerfIncreaseThreshold](#)

[PerfIncreaseTime](#)

[PerfDecreaseThreshold](#)

[PerfDecreaseTime](#)

[PerfLatencyHint](#)

[PerfAutonomousMode](#)

[PerfEnergyPreference](#)

[PerfAutonomousWindow](#)

## DutyCycling

Static configuration options for heterogeneous power scheduling

HeteroIncreaseThreshold

HeteroDecreaseThreshold

HeteroIncreaseTime

HeteroDecreaseTime

HeteroClass1InitialPerf

HeteroClass0FloorPerf

## Battery settings

Critical battery action

Critical battery threshold

Low battery action

Low battery threshold

Low battery warning

Reserve battery level

## Power button and lid settings

Lid open wake action

Lid switch close action

Power button action

Power button forced shutdown

Sleep button action

## Display settings

Adaptive display idle timeout

Allow display required policy

Dim annoyance timeout

Dim display brightness

Display brightness level

Display idle timeout

## Disk settings

Disk burst ignore time

Disk idle timeout

Link power management mode - adaptive

- [Link power management mode - HIPM/DIPM](#)
- [Energy Saver settings](#)
  - [Battery threshold](#)
  - [Brightness](#)
- [PCI Express settings](#)
  - [Link state power management](#)
- [Sleep settings](#)
  - [Allow away mode](#)
  - [Allow sleep with open remote files](#)
  - [Allow sleep states](#)
  - [Allow system required requests](#)
  - [Automatically wake for tasks](#)
  - [Hibernate idle timeout](#)
  - [Hybrid sleep](#)
  - [Sleep idle timeout](#)
  - [Sleep unattended idle timeout](#)
- [Other power settings](#)
  - [Device idle policy](#)
  - [Prompt for password on resume](#)
  - [Allow networking during standby](#)
- [Legacy configuration options](#)
  - [PERFBOOSTMODE](#)
  - [PERFBOOSTPOL](#)
- [Preinstalled and exclusive apps](#)
  - [Exclusive apps](#)
    - [Preinstallable apps for desktop devices](#)
    - [Preinstallable apps for mobile devices](#)
  - [Preinstall tasks](#)
- [Change history for customization docs](#)

# Customize

10/2/2018 • 2 minutes to read • [Edit Online](#)

## Purpose

Customizations of the Windows OS are ways in which partners can modify the Windows device UI, connectivity settings, and user experience to better reflect the partners' brand, and to fit the network and market in which the device ships. Customization options include adding applications, modifying icons and Start layouts, configuring network settings by using device management, changing defaults in **Settings**, and adding brand-specific art and sounds to the OS.

Windows 10 supports both pre-existing desktop Unattend settings, and mobile Managed Centralized Settings Framework (MCSF), for configuring customization options for Windows 10 devices.

See the following sections for more information about what you can do to customize your Windows 10 devices.

Topic	Description
<a href="#">Customizations for desktop</a>	This section includes topics describing key desktop customization opportunities, as well the <a href="#">Unattended Windows Setup Reference</a> , and <a href="#">Windows System Image Manager Technical Reference</a> .
<a href="#">Customizations for enterprise desktop</a>	Learn about the customizations available to you if you are providing a controlled and specialized experience on a Windows device running Windows 10 Enterprise.
<a href="#">Customizations for mobile</a>	Learn about the customizations for mobile enterprise, which allow you to run mobile line-of-business applications on a platform that ensures that data is captured securely and efficiently. This section includes all customization options available as part of the Managed Centralized Settings Framework (MCSF).
<a href="#">Configure power settings</a>	Learn about the power settings you can configure using the Windows provisioning framework. Each power setting topic includes the identification GUID, allowed values, meaning, and common usage scenarios for the setting.
<a href="#">Preinstalled and exclusive apps</a>	If you're a Windows OEM or mobile operator partner, find out how you can create partner apps that you can package and configure to install during the initial device setup process. While the user is going through the initial setup process, the preinstalled apps are installed in the background. OEMs can also work with software developers to target OEM devices for apps to appear exclusively on, based on registry keys.
<a href="#">Change history for Customize</a>	Review the timeline of Windows 10 Customization topics that have been created, updated, or deleted.

## Audience

This section of the partner documentation is intended for Original Equipment Manufacturers (OEMs), Original Design Manufacturers (ODMs), Independent Hardware Vendors (IHVs), system builders, mobile operators, and IT

administrators.

If you have purchased a Windows 10 device and would like to learn more about using its features, please see Microsoft's Windows support site online at <https://support.microsoft.com/en-us/products/windows?os=windows-10>.

# Customizations for desktop devices

10/8/2018 • 2 minutes to read • [Edit Online](#)

You have the following options to customize your image. Depending on which options you'd like to use, you'll employ the associated method or choice of methods to apply the customization.

FEATURE	UNATTEND	MODIFICATION FILE
Taskbar	subset	TaskbarLayoutModification.xml
Start layout	subset	LayoutModification.xml
Out of Box Experience (OOBE)	subset	OOBE.xml
Darkmode	yes	Unattend.xml
Get Help app	yes	Unattend.xml
Colors	yes	Unattend.xml

## NOTE

All desktop customization options listed above are supported in Windows 10 in S mode. To learn more, see [Windows 10 in S mode manufacturing overview](#).

## In this section

These are some common ways to customize your desktop device. You will also find the technical reference for Unattend and WSIM.

TOPIC	DESCRIPTION
<a href="#">Customize the taskbar</a>	You can pin up to three additional apps to the taskbar by adding a taskbar layout modification file, for example, TaskbarLayoutModification.xml. You can specify different taskbar configurations based on SKU, device locale, or region.
<a href="#">Customize the Start layout</a>	Learn how to customize the size of the start menu, and add your own tiles to it.
<a href="#">Customize OOBE</a>	When customers turn on their Windows PCs for the first time, they will see the Windows Out of Box Experience (OOBE). Customize OOBE to determine how much work customers must do to complete the OOBE screens before they can enjoy their PCs running Windows 10.
<a href="#">Customize the Retail Demo Experience (RDX)</a>	Showcase your new devices on the retail sales floor with a rich, engaging videos and experiences.

Topic	Description
<a href="#">Customize the Windows power slider</a>	The Windows Performance Power slider enables end customers to quickly and intelligently trade performance of their system for longer battery life. You can set the default slider mode for both AC and DC, and configure the power settings and PPM options that are engaged in each power slider mode.
<a href="#">Set dark mode</a>	This personalization setting for end users allows them to express preference whether to see applications which support the setting in a dark or light mode. You can set the dark mode as the default for apps using Unattend.
<a href="#">Customize the Get Help app</a>	The Get Help app empowers customers to self-help with troubleshooters, instant answers, Microsoft support articles, and more, before contacting assisted support. You can customize the Get Help app to surface your support app or support website.
<a href="#">Customize SIM card slot names</a>	You can customize the names of SIM card slots on the device to more easily differentiate between them. For example, if the device has both an embedded SIM slot and an external SIM slot, customizing the names will help your customers understand which is which.
<a href="#">Customize a Specific Absorption Rate mapping table</a>	You can configure and store a Specific Absorption Rate (SAR) table for mobile broadband modems in the registry. When a mobile broadband modem is connected to the Windows device, Windows automatically uses the table to map the mobile country code (MCC) of the modem's registered mobile operator (MO) to its appropriate SAR back-off index, and configures the modem with it.
<a href="#">Pen and Windows ink</a>	You can create an advanced Pen settings app, or link to your own apps, in the Pen and Windows Ink settings.
<a href="#">Windows SIM Technical Reference</a>	Settings reference for Windows System Image Manager.
<a href="#">Unattended Windows Setup Reference</a>	Settings reference for Unattend.

## Related topics

[OEM deployment of Windows 10 for desktop editions](#)

[Planning a Windows 10 in S mode deployment](#)

[Deployment options](#)

# Customize the Taskbar

10/8/2018 • 4 minutes to read • [Edit Online](#)

You can pin up to three additional apps to the taskbar. There are two methods to do this:

- **Taskbar Layout Modification XML** method (recommended)
  - Supports multivariant images; you can specify different sets of taskbar layouts for different regions.
  - Uses a single XML file.
  - Is the only method that allows you to add UWP apps to the taskbar.
  - In the examples below, the file name “TaskbarLayoutModification.xml” is used, however, you can choose any name you like.
- **Classic Unattend method** (still supported in Windows 10, but marked as deprecated, and may not be available in future builds)
  - Uses the Unattend setting: [TaskbarLinks](#)

## Taskbar links and ordering

The taskbar starts with the following links: **Start**, **Search**, and **Task View**, plus four additional Windows-provided links: Mail, Edge, File Explorer, and Store. These pins cannot be removed or replaced.

OEMs can add up to three additional pinned apps to the taskbar.

For left-to-right languages, the taskbar icons are ordered from left to right (Start, Search, Task View, Windows-provided Pins, OEM-provided pins, Mail). For right-to-left languages, the taskbar icons are in the opposite order, with the right-most element being **Start**.

## Add a default path

To use a Taskbar Layout Modification XML file in Windows, you'll need to add a registry key (LayoutXMLPath) to the image, and then generalize and recapture the image. The registry key must be processed before the specialize configuration pass. This means you won't be able to simply add the registry key by using Synchronous Commands/FirstLogonCommands unless you plan to generalize the image afterwards.

You can use any name or file location by defining this in the registry key; the filename and path to TaskbarLayoutModification.xml is not required. The other shortcut files, apps, and the Taskbar Layout Modification file itself can be changed at any time through regular imaging techniques. You can add this registry key to all your images, even if you intend to add taskbar links using the Classic Unattend method.

## Configure taskbarlayoutmodification.xml

1. Install the Windows image to a technician computer.
2. After the image boots, go into audit mode by pressing CTRL+SHIFT+F3.
3. Add the following registry key to define a default location for the Taskbar Layout Modification file:

```
cmd /c reg add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\ /v LayoutXMLPath /d C:\Windows\Fabrikam\TaskbarLayoutModification.xml
```

4. Add a Taskbar Layout Modification file (TaskbarLayoutModification.xml) in the default location for example:

```
C:\Windows\Fabrikam\TaskbarLayoutModification.xml
```

```

<?xml version="1.0" encoding="utf-8"?>
<LayoutModificationTemplate
    xmlns="http://schemas.microsoft.com/Start/2014/LayoutModification"
    xmlns:defaultlayout="http://schemas.microsoft.com/Start/2014/FullDefaultLayout"
    xmlns:start="http://schemas.microsoft.com/Start/2014/StartLayout"
    xmlns:taskbar="http://schemas.microsoft.com/Start/2014/TaskbarLayout"
    Version="1">

    <CustomTaskbarLayoutCollection PinListPlacement="Replace">
        <defaultlayout:TaskbarLayout>
            <taskbar:TaskbarPinList>
                <taskbar:UWA AppUserModelID="Microsoft.Windows.Photos_8wekyb3d8bbwe!App" />
                <taskbar:DesktopApp DesktopApplicationLinkPath="%ALLUSERSPROFILE%\Microsoft\Windows\Start
Menu\Programs\Accessories\Paint.lnk"/>
            </taskbar:TaskbarPinList>
        </defaultlayout:TaskbarLayout>
        <defaultlayout:TaskbarLayout Region="US|GB">
            <taskbar:TaskbarPinList>
                <taskbar:DesktopApp DesktopApplicationLinkPath="%APPDATA%\Microsoft\Windows\Start
Menu\Programs\Accessories\Notepad.lnk" />
                <taskbar:UWA AppUserModelID="Microsoft.WindowsCalculator_8wekyb3d8bbwe!App" />
            </taskbar:TaskbarPinList>
        </defaultlayout:TaskbarLayout>
        <defaultlayout:TaskbarLayout Region="CN|TW">
            <taskbar:TaskbarPinList>
                <taskbar:DesktopApp DesktopApplicationLinkPath="%APPDATA%\Microsoft\Windows\Start
Menu\Programs\Accessories\Notepad.lnk" />
                <taskbar:UWA AppUserModelID="Microsoft.Windows.Photos_8wekyb3d8bbwe!App" />
                <taskbar:DesktopApp DesktopApplicationLinkPath="%ALLUSERSPROFILE%\Microsoft\Windows\Start
Menu\Programs\Accessories\Paint.lnk"/>
            </taskbar:TaskbarPinList>
        </defaultlayout:TaskbarLayout>
    </CustomTaskbarLayoutCollection>
</LayoutModificationTemplate>

```

5. Generalize the Windows image using [Sysprep](#):

```
Sysprep /generalize /oobe /shutdown
```

6. Boot to Windows PE.

7. Recapture the image. For example:

```
Dism /Capture-Image /CaptureDir:C:\ /ImageFile:c:\install-with-new-taskbar-layout.wim /Name:"Windows
image with Taskbar layout"
```

8. You can now apply this image to other PCs.

### To reference your apps

- For **Classic Windows applications**, use shortcut (.lnk) files. We recommend using the same shortcut .lnk files in the All Users Start menu. Example:

```

DesktopApp
DesktopApplicationLinkPath="%ALLUSERSPROFILE%\Microsoft\Windows\Start
Menu\Programs\Accessories\Paint.lnk"

```

- For **Universal Windows apps**, use the Universal Windows app user model ID. Example:

```
UWA AppUserModelID="Microsoft.Windows.Photos_8wekyb3d8bbwe!App"
```

#### NOTE

Links to .url files are not supported.

### To use different layouts for different regions

To use different layouts for different regions, include a region in the defaultlayout tag. These regions use the second half of the language/region tags listed in [Available Language Packs for Windows](#). You can use multiple region tags separated by a pipe (|) character. Here is an example of adding pins to the Chinese (PRC) and Chinese (Taiwan) regions:

```
<defaultlayout:TaskbarLayout Region="CN|TW">
```

## How Windows parses the setting for Unattend and Taskbar Layout Modification XML

While you're transitioning to the new method to customize the taskbar, you may end up using existing images that still include your old Unattend TaskbarLinks settings. When that happens:

1. If Windows finds a valid Taskbar Layout Modification XML file, it uses the XML file, and ignores any of the Unattend taskbar settings.
2. If the Taskbar Layout Modification XML file isn't found, or is invalid, Windows looks for the old Unattend TaskbarLinks settings. If it finds them, it uses them.
3. If Windows can't find either a valid Taskbar Layout Modification XML file, or Unattend TaskbarLink settings, then only the Windows-provided pins and **Start**, **Search**, and **Task View** are shown.

### Set transparency for the taskbar

The default transparency setting for the taskbar is 15%. To make Taskbar work with the Dark Mode on OLED displays, you need to set the taskbar transparency to 40%.

To set the transparency for the Taskbar, create a registry key called "UseOLEDTaskbarTransparency" and place it in the following location:

```
HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced
```

#### IMPORTANT

This registry key should only be used to change the taskbar transparency for OLED screens. We do not advise changing the default transparency on non-OLED displays.

## Action Center

Most quick action tiles that are pinned in the Action Center are not customizable. You can, however, enable one of the desktop quick action tiles, **Color Profile**, if more than one color profile is installed on the device. To let users see Color Profile in the action center:

1. Install at least two ICC color profiles on the primary display. For more information on how to accomplish this, please work with your Microsoft representative.
2. Add the following registry key to enable the Microsoft.QuickAction.ColorProfile quick action:

```
HKLM\Software\Microsoft\Shell\OEM\QuickActions\ColorProfileQuickAction = 1
```

3. Add the following registry key to display the Microsoft.QuickAction.ColorProfile quick action.

```
HKLM\Software\Microsoft\Shell\OEM\QuickActions\Microsoft.QuickAction.ColorProfile
```

# Customize the Start layout

10/2/2018 • 11 minutes to read • [Edit Online](#)

You can customize the Start layout by creating a `LayoutModification.xml` file and configuring the settings. To determine the overall look of the Start layout, the default layout is applied based on SKU and region, and then the `LayoutModification.xml` or `Unattend.xml` file is processed.

You can customize the following aspects of the Start layout:

- The size, including the number of columns and number of tiles per row
- The tiles in both OEM Groups; including the size, position, and the app or web link associated with each tile
- The display layout for the Microsoft Office suite of tiles
- Create Start layouts for each region you support

After customizing the Start layout, use Windows Configuration Designer to add the file to the device image. See [Add the LayoutModification.xml file to the device](#) for instructions.

## LayoutModification.xml

The XML schema for `LayoutModification.xml` requires the following order for tags directly under the `LayoutModificationTemplate` node:

1. `LayoutOptions`
2. `DefaultLayoutOverride`
3. `RequiredStartGroupsCollection`
4. `AppendDownloadOfficeTile` –OR– `AppendOfficeSuite` (only one Office option can be used at a time)
5. `AppendOfficeSuiteChoice`
6. `TopMFUApps`
7. `CustomTaskbarLayoutCollection`
8. `InkWorkspaceTopApps`

Comments are not supported in the `LayoutModification.xml` file.

For an inclusive list of settings you can configure in `LayoutModification.xml`, a full XML example, and instructions on adding the XML file to the device, see [Start layout XML for desktop editions of Windows 10 \(Reference\)](#)

## Customize the size of the Start layout

We recommend that you set the default Start layout so it is not greater than 40% of the size of the desktop. If it is greater than half the width of the desktop, customers might perceive that the device and Windows are optimized only for touch, and feel less satisfied when they use a mouse and keyboard.

Use `LayoutOptions` in `LayoutModification.xml` to indicate the number of columns, and the number of tiles per row, in the Start layout.

### Specify the number of columns in the Start layout

You have three options for the Start layout size: **small** (one column of tiles), **medium** (2 columns of tiles), or **full screen**. New devices running Windows for desktop will default to a Start layout with two columns of tiles unless boot to tablet mode is enabled. Devices with screens that are under 10" have boot to tablet mode enabled by default. For these devices, users see the **full screen** Start layout on the desktop.

## NOTE

We suggest you leave the default values for these features so that Windows can use its own logic to do the right thing for the customer. You can, however, adjust the following OS features if you have a scenario that requires it. For example, if you have a device that is meant mainly for use as a tablet, but is bigger than 10", you can use [SignInMode](#).

Here is how you set the size of the start layout, using LayoutModification.xml.

- To set as small, with one column of tiles:

```
<LayoutOptions  
    StartTileGroupsColumnCount="1"  
    FullScreenStart="false"  
/>
```

- To set as medium, with two columns of tiles:

```
<LayoutOptions  
    StartTileGroupsColumnCount="2"  
    FullScreenStart="false"  
/>
```

- To set as full screen (and set the default to one column if the user disables full screen):

```
<LayoutOptions  
    StartTileGroupsColumnCount="1"  
    FullScreenStart="true"  
/>
```

## IMPORTANT

Setting `FullScreenStart` to true requires rebooting the device to take effect.

## Specify the number of tiles per row in the Start layout

You can configure your Start layout to show either 6 or 8 medium tiles per row using `StartTileGroupCellWidth` in `LayoutModification.xml`.

We recommend you configure this setting to optimize the Start layout for the size of your device's screen. If this setting is not configured in `LayoutModification.xml`, Windows will use its own logic to set the number of tiles per row.

```
<LayoutModificationTemplate  
    xmlns="http://schemas.microsoft.com/Start/2014/LayoutModification"  
    xmlns:defaultlayout="http://schemas.microsoft.com/Start/2014/FullDefaultLayout"  
    xmlns:start="http://schemas.microsoft.com/Start/2014/StartLayout"  
    Version="1">  
    <LayoutOptions StartTileGroupCellWidth="8" />  
</LayoutModificationTemplate>
```

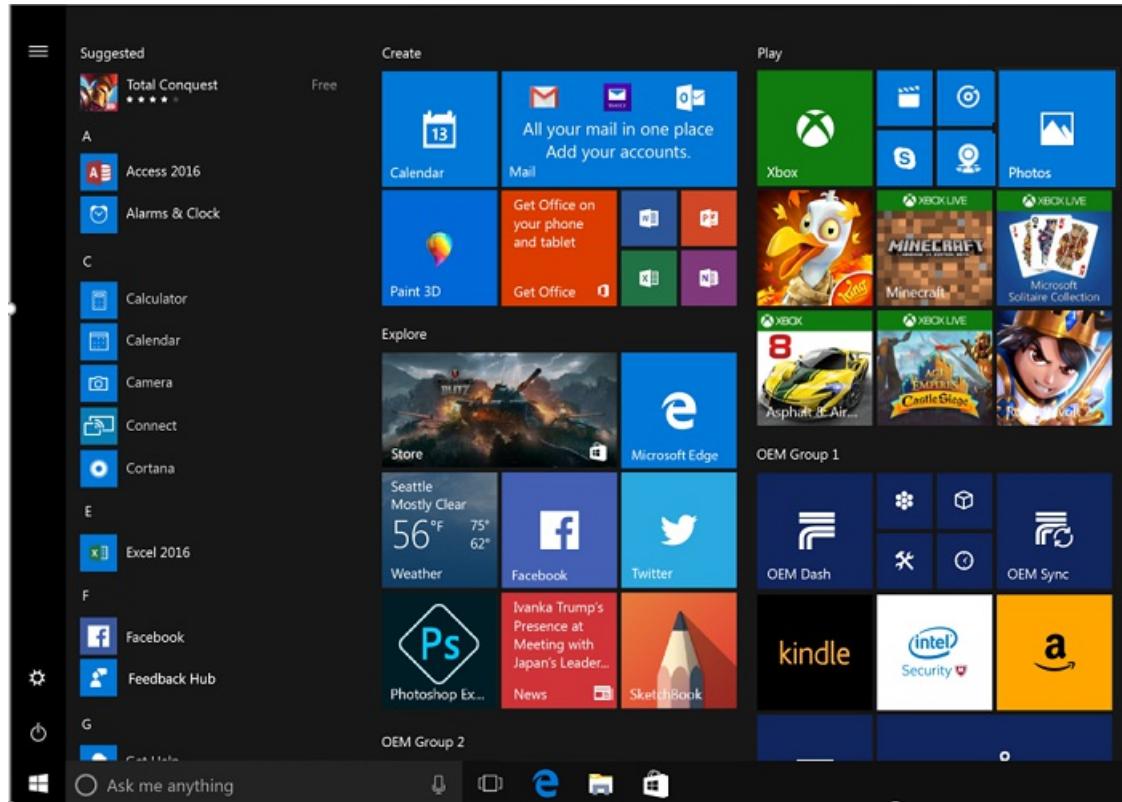
The number of tiles you select will be divided evenly between the two columns in the Start layout. For example, if you choose to show 6 medium tiles per row, each columns will contain 3 tiles.

## Customize OEM Groups in the Start layout

You can pin tiles in up to two OEM groups. You'll specify the OEM groups, and tiles they contain, in `LayoutModification.xml`. The first group you specify in `LayoutModification.xml` will be placed in the first open, available part of the Start layout, scanning top to bottom. If you've specified a second OEM group in `LayoutModification.xml`, it will be placed after the first group, in the first open, available part of the Start layout, scanning top to bottom.

If you have `StartTileGroupCellWidth` set to 6 in `LayoutModification.xml`, the OEM group will be 3 medium tiles wide by 3 high. If you have `StartTileGroupCellWidth` set to 8, the OEM group will be 4 medium tiles wide (group one is 3 tiles high by 4 wide, and group two is 2 high by 4 wide).

Here is an example that shows a Start layout with two OEM groups:



In the example above, `StartTileGroupCellWidth` is set to 6, and `StartTileGroupsColumnCount` is set to 2. Subsequently, the Start layout shows 2 columns of tiles, and each row is 6 medium tiles long.

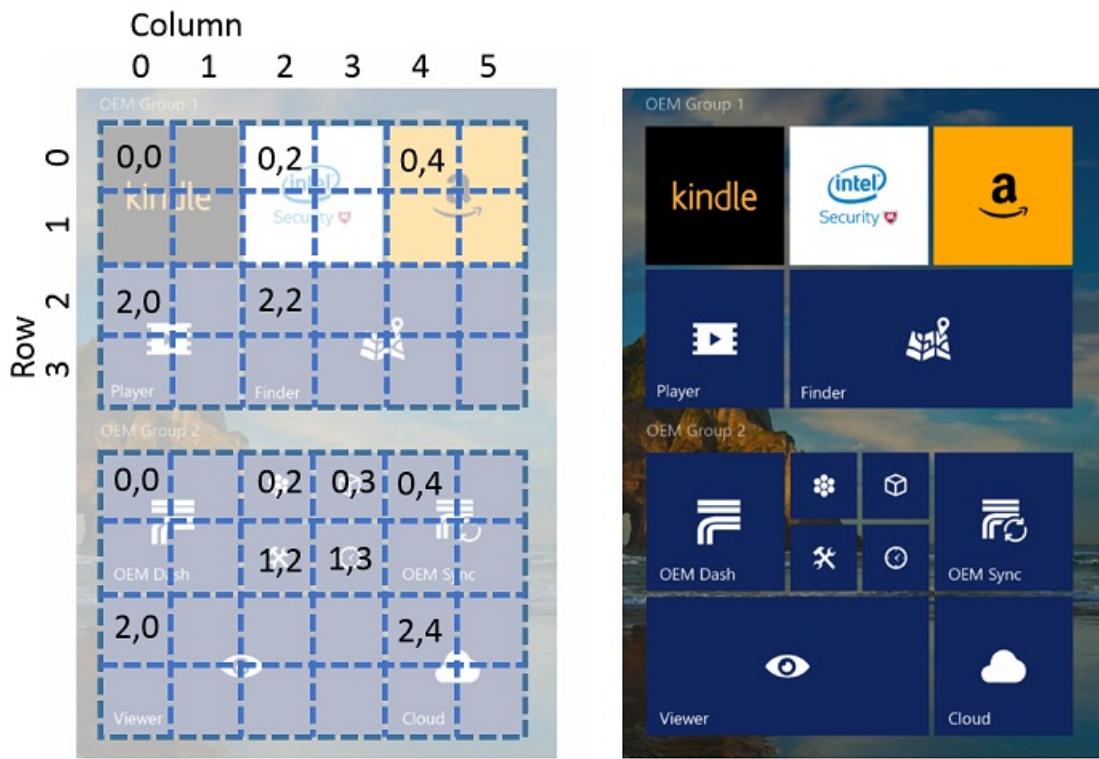
**Customize the size and position of tiles in the OEM Groups**

The `LayoutModification.xml` file allows four tile size options for each tile you add to your OEM groups. The available tile sizes are:

- Small tile: 1x1
  - Medium square tile: 2x2
  - Wide tile: 4x2
  - Large tile: 4x4

The `Row` and `Column` elements determine the position of the upper, left edge of the tile within the group. The `0,0` position is the first row, first column.

For example, here is a Start layout with the row and column grid overlaid.



## Add tiles that launch apps and web links

You can configure each of your tiles to launch:

- A Universal Windows app (using the `start:Tile` tag)
- A Windows 8 or 8.1 app (using the `start:Tile` tag)
- A desktop application (using the `start:DesktopApplicationTile` tag)
- A web link that opens in the default browser (using the `start:DesktopApplicationTile` tag)
- A web link that opens in Edge (using the `start:SecondaryTile` tag)

### NOTE

Each tile pinned to the Start layout can launch a single UWP app, Microsoft Store app, desktop app, or web link. A tile can't be a group of apps or a folder.

### App tiles

You can add an app tile that will launch a Universal Windows app, or a Windows 8/8.1 app, using `start:Tile` in `LayoutModification.xml`. To specify the app you wish to launch, you must set the `AppUserModelID` attribute of `start:Tile` to the application user model ID (AUMID) associated with the app. The AUMID is case-sensitive.

### IMPORTANT

In Windows 10, version 1803, all apps must either be pinned to the Start layout, and/or pre-installed using the new `region` parameter in DISM, otherwise they will be removed on any system that uses that layout. See [Preinstall apps using DISM](#) for guidance on using the new parameter.

The following example shows how to pin the Microsoft Edge Universal Windows app:

```
<start:Tile  
    AppUserModelID="Microsoft.MicrosoftEdge_8wekyb3d8bbwe!MicrosoftEdge"  
    Size="2x2"  
    Row="0"  
    Column="0"/>
```

You can use the `start:DesktopApplicationTile` tag to pin a Windows desktop application to Start. There are two ways you can specify a Windows desktop application:

- By setting `DesktopApplicationLinkPath` to a path to a shortcut link (.lnk file) to a Windows desktop application.
- By setting the `DesktopApplicationID` to the application's ID, if this is known. If the Windows desktop application doesn't have one, use the shortcut link option.

The following example shows how to pin the Command Prompt desktop application using the .lnk method:

```
<start:DesktopApplicationTile  
    DesktopApplicationLinkPath="%appdata%\Microsoft\Windows\Start Menu\Programs\System Tools\Command Prompt.lnk"  
    Size="2x2"  
    Row="0"  
    Column="4"/>
```

The following example show how to pin the File Explorer Windows desktop application by specifying the desktop application ID:

```
<start:DesktopApplicationTile  
    DesktopApplicationID="Microsoft.Windows.Explorer"  
    Size="2x2"  
    Row="0"  
    Column="2"/>
```

#### Web link tiles

You can add a web link tile that will open in the default browser, or you can add a *secondary tile* that will specifically open in Microsoft Edge. To create a web link tile that will open in the default browser, create a .url file:

1. Right click on Desktop > New > Shortcut
2. Type a URL such as <http://www.fabrikam.com>
3. Click Next
4. Type a name for the shortcut such as Fabrikam and click Finish. The .url file is saved to your desktop.
5. Add the .url file to the image in the `%ALLUSERSPROFILE%\Microsoft\Windows\Start Menu\Programs\` folder , and then add a `DesktopApplicationTile` element to `LayoutModification.xml` :

```
<!-- Web link tile with associated .url file in StartMenu folder -->  
<start:DesktopApplicationTile  
    DesktopApplicationID="www.Fabrikam.com"  
    Size="2x2"  
    Row="0"  
    Column="2"/>
```

To create a secondary tile (a web link tile that will open in Microsoft Edge), add a `SecondaryTile` element to `LayoutModification.xml` and specify Edge in the `AppUserModelID` attribute.

```

<!-- Web link tile that launches in Edge -->
<start:SecondaryTile
    AppUserModelID="Microsoft.MicrosoftEdge_8wekyb3d8bbwe!MicrosoftEdge"
    TileID="MyWeblinkTile"
    Arguments="http://msn.com"
    DisplayName="MySite"
    Square150x150LogoUri="ms-appx:///Assets/MicrosoftEdgeSquare150x150.png"
    Wide310x150LogoUri="ms-appx:///Assets/MicrosoftEdgeWide310x150.png"
    ShowNameOnSquare150x150Logo="true"
    ShowNameOnWide310x150Logo="false"
    BackgroundColor="#FF112233"
    Size="2x2"
    Row="0"
    Column="4"/>

```

#### **NOTE**

The Edge tile itself cannot be customized. The icon, text and the page that it launches must remain the default.

The OEM-custom icon and supporting text in the tile must:

- Logically relate to the activity or action the user is expected to take
- Launch the Edge browser
- The icon and supporting text should not imply that it is anything other than Edge (e.g. another browser)
- The page that it goes to must work in Edge
- The page should not promote another browser

```

<start:SecondaryTile AppUserModelID="Microsoft.MicrosoftEdge_8wekyb3d8bbwe!MicrosoftEdge"
    TileID="MyWeblinkTile"
    Arguments="http://msn.com"
    DisplayName="MySite"
    Square150x150LogoUri="ms-appx:///Assets/MicrosoftEdgeSquare150x150.png"
    Size="2x2"
    Row="0"
    Column="4"/>

```

## Customize the Office suite of tiles

The Microsoft Office suite of tiles is the first group of tiles in the Start layout. There are a few different options available to customize this suite of tiles.

- If you've pre-installed Office Desktop Bridge to the device, use the `AppendOfficeSuite` and `AppendOfficeSuiteChoice` tags in `LayoutModification.xml`.
- If you've pre-installed Office Mobile to the device, use only the `AppendOfficeSuite` tag in `LayoutModification.xml`.
- If you have not pre-installed Office to the device, you can use the `AppendDownloadOfficeTile` tag in `LayoutModification.xml` to add a **Download Office** tile to the suite.

#### **NOTE**

The version of Office you indicate in `LayoutModification.xml` must match the version of Office that's pre-installed to the device.

## Office Desktop Bridge

We recommend pre-installing Office Desktop Bridge on all devices where the screen is 10.1 inches or larger. Office Desktop Bridge is included in the OEM Pre-installation Kit (OPK). After pre-installing Office Desktop Bridge, each Office app appears as a tile in the Office suite of tiles.

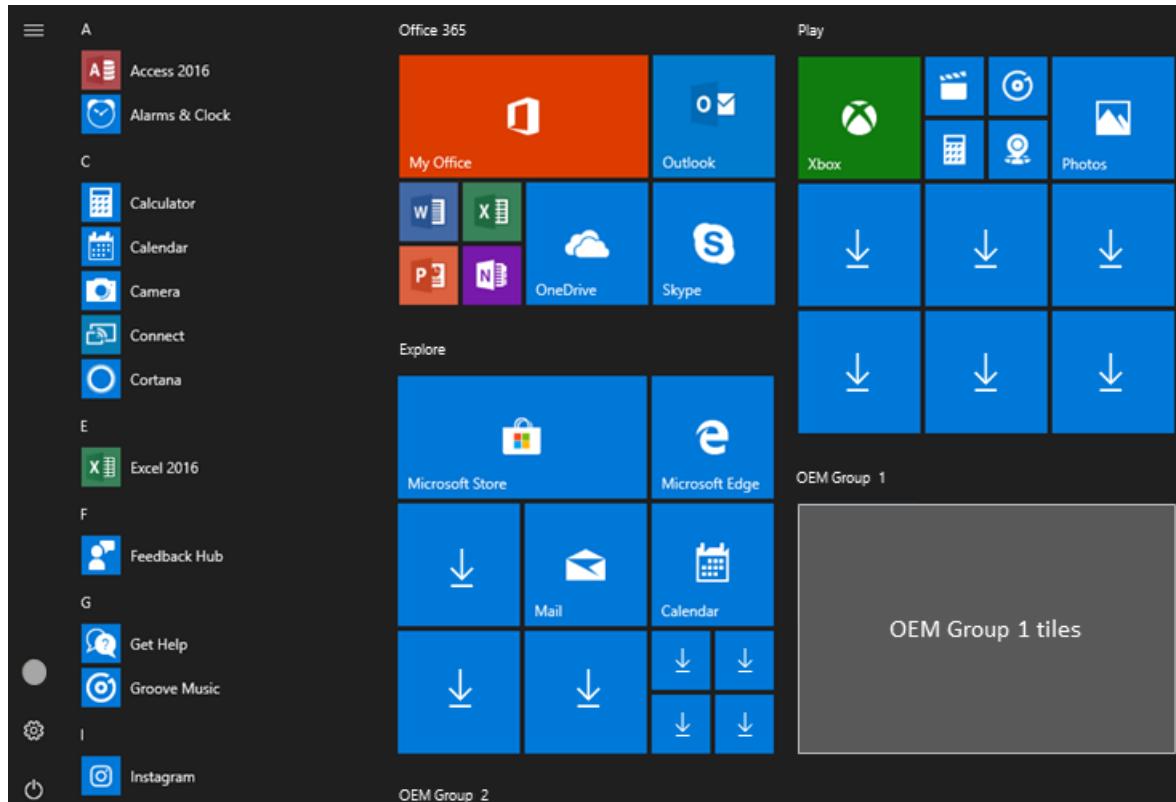
In Windows 10, version 1803, add the following two tags to `LayoutModification.xml`:

- `<AppendOfficeSuite/>`
- `<AppendOfficeSuiteChoice Choice="DesktopBridgeSubscription"/>`

For example:

```
<LayoutModificationTemplate
    xmlns="http://schemas.microsoft.com/Start/2014/LayoutModification"
    xmlns:defaultlayout="http://schemas.microsoft.com/Start/2014/FullDefaultLayout"
    xmlns:start="http://schemas.microsoft.com/Start/2014/StartLayout"
    Version="1">
    <AppendOfficeSuite/>
    <AppendOfficeSuiteChoice Choice="DesktopBridgeSubscription"/>
</LayoutModificationTemplate>
```

This will set the heading of the Office suite of tiles to **Office 365**, to highlight the Office 365 apps you've made available on the device. For example:



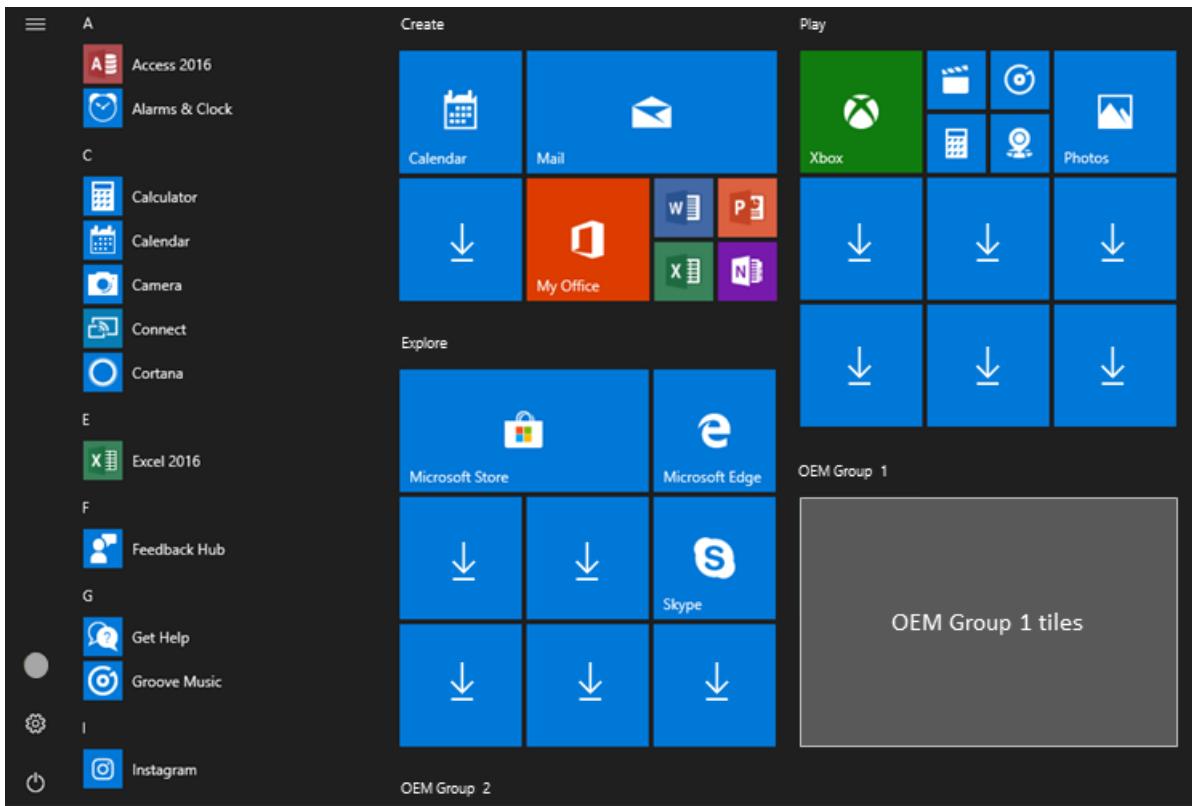
#### NOTE

Tile sizes and positions may vary based on the device SKU, region, and the size of the Start layout.

For older versions of Windows, and for devices shipping with Activation for Office (AFO) Perpetual, add the following two tags to `LayoutModification.xml`:

- `<AppendOfficeSuite/>`
- `<AppendOfficeSuiteChoice Choice="DesktopBridge"/>`

This will set the heading to **Create**. For example:



We advise using these tags when AFO Perpetual (Office 2016) is pre-installed to provide the best user experience for your customers. The new Office block of tiles is labeled **Office 365**, which is not the same as Office 2016, and could be confusing to users.

## Office Mobile

We recommend pre-installing Office Mobile apps on all devices where the screen size is smaller than 10.1 inches.

After you've pre-installed Office Mobile apps to the device, use only the `<AppendOfficeSuite>` tag in `LayoutModification.xml` to configure the Start layout. For example:

```

<LayoutModificationTemplate
    xmlns="http://schemas.microsoft.com/Start/2014/LayoutModification"
    xmlns:defaultlayout="http://schemas.microsoft.com/Start/2014/FullDefaultLayout"
    xmlns:start="http://schemas.microsoft.com/Start/2014/StartLayout"
    Version="1">
    <AppendOfficeSuite/>
</LayoutModificationTemplate>

```

The Office mobile apps will appear as tiles in the Start layout under the heading **Create**.

## Download Office

If you have not pre-installed Office to the device, you can append the **Download Office** tile to Start. This replaces the **My Office** tile that appears in the middle of the second row with the classic desktop app download tile, and supports all OEM scenarios including Activation for Office (AFO) and Pre-install PC (PIPC).

To append the **Download Office** tile, add the `<AppendDownloadOfficeTile/>` tag in your `LayoutModification.xml` file. For example:

```
<LayoutModificationTemplate  
    xmlns="http://schemas.microsoft.com/Start/2014/LayoutModification"  
    xmlns:defaultlayout="http://schemas.microsoft.com/Start/2014/FullDefaultLayout"  
    xmlns:start="http://schemas.microsoft.com/Start/2014/StartLayout"  
    Version="1">  
    <AppendDownloadOfficeTile/>  
</LayoutModificationTemplate>
```

## Create Start layouts for each region you support

You can use the `Region` parameter of the `RequiredStartGroups` tag in your `LayoutModification.xml` file to specify Start layouts per region. To learn more, see [RequiredStartGroups tag](#) in the Start Layout XML Reference.

Alternately, you can use multivariant capabilities in Windows provisioning to create different Start layouts per region. To learn more, see [Use Windows Provisioning multivariant support](#) in the Start Layout XML Reference.

## First run tasks

First Run Tasks are background tasks that are active when the user first signs into Windows. First Run Tasks are not available in `LayoutModification.xml`. However, you can still use them by including an `Unattend.xml` file with `StartTiles` tags using the same AppID as in `LayoutModification.xml`.

If the `AppendGroup` tag is present in `LayoutModification.xml`, it will override `Unattend.xml` for all Start pinning. However, if an `Unattend.xml` `StartTiles` tag exists for the same AppID as in `LayoutModification.xml`, the `FirstRunTask` from `Unattend.xml` will be respected.

For example, include a `LayoutModfication.xml` file specifying an app like this:

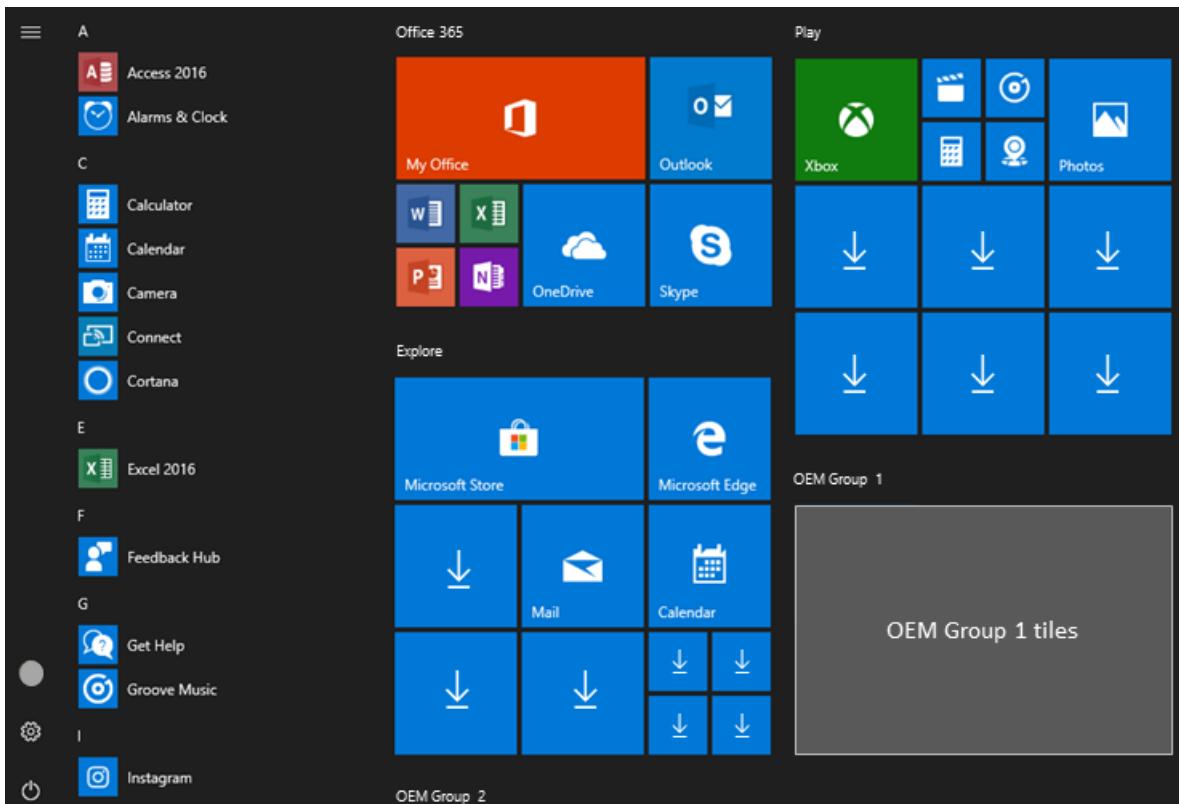
```
<start:Tile AppUserModelID="Microsoft.MicrosoftEdge_8wekyb3d8bbwe!MicrosoftEdge" Size="2x2" Row="0"  
Column="0"/>
```

Also include an `Unattend.xml` file specifying the same AppID like this:

```
<SquareOrDesktopTile5>  
    <AppId>Microsoft.MicrosoftEdge_8wekyb3d8bbwe!MicrosoftEdge</AppId>  
    <FirstRunTask>BackgroundTasks_Notifications.Services.MessagingBackgroundTask</FirstRunTask>  
</SquareOrDesktopTile5>
```

## Dynamically delivered apps

Some apps on the Start layout are downloaded dynamically after the Out of Box Experience (OOBE) completes. If the device is on a metered network, or without network connectivity, app downloads are paused, and the user will see down arrows instead of the app name on the app tiles, as in the following image.



The downloads start or resume after the network connects.

## Related topics

- [Start layout XML for desktop editions of Windows \(Reference\)](#)
- [Add the LayoutModification.xml file to the device](#)
- [StartTiles Unattend setting](#)

# Customize the Out of Box Experience (OOBE)

10/2/2018 • 5 minutes to read • [Edit Online](#)

When customers turn on their Windows PCs for the first time, they will see the Windows Out of Box Experience (OOBE). OOBE consists of a series of screens that require customers to accept the license agreement, connect to the internet, log in with, or sign up for a Microsoft Account, and share information with the OEM.

During OOBE, Cortana voice-over strings will assist users by setting the context of each screen, and requesting their input. While voice assistance is more accessible to the non-sighted, the design is focused at being inclusive to all our customers. Cortana voice is intended to be novel and supplementary to increase user engagement in all places in OOBE. Cortana voice also helps reduce cognitive load by offering informationally-identical, but differently-phrased information. We still expect non-sighted users to enable screen readers to get through OOBE. Some pages in OOBE do not accept voice input, and instead require a keyboard or mouse to complete the action. Cortana voice will clearly communicate input requirements (voice or keyboard/mouse) to the user.

## TIP

We recommend you target a 65 decibel peak volume during OOBE. To test for this volume, measure an audio sample from 2 feet (60 centimeters) away from the device.

The OOBE flow is also designed to reduce cognitive load significantly by breaking up tasks into discrete chunks. Although there are several pages in the OOBE flow, each one requests a specific action or input from the user. This is helpful for our average customer (and even many power users) and has shown to reduce fatigue significantly.

## OOBE flow

The following is a non-exhaustive list of screens the user may see during OOBE, in order:

1. **Language selection**
2. **Cortana welcome**
3. **Region selection**
4. **Keyboard selection**
5. **Connect to a network**
6. **Automatic download of critical ZDP and driver updates.** See [Windows updates during OOBE](#) for more details.
7. **Get the latest from Windows.** Prior to Windows 10, version 1803, this screen was named **Your PC has an update waiting** and it appeared at the end of OOBE.
8. **End User License Agreement (EULA)**
9. **Sign in to, or create, a local account or Microsoft account (MSA).** If a user chooses the local account option, the **Sign in with Microsoft instead?** screen will appear next in the OOBE flow. This screen encourages the user to sign in with their MSA for an optimal Windows experience.
10. **Create security questions for a local account.** New in Windows 10, version 1803. Only displays if the user chose to create a local account, rather than logging into their MSA, on the previous screen. See [OOBE screen details](#) to learn more about this new screen in OOBE.
11. **Windows Hello setup**
12. **Link your phone and PC.** This screen will only appear if the user signed into their Microsoft account, and connected to a network, on the previous screens.
13. **Save files to OneDrive.** This is a cloud service page.

14. **Set up Office.** This screen is only displayed if the user is connected to a network, and has provided their Microsoft account information. Content on the page will vary depending on the user's account type. For example, if their Microsoft account qualifies for a free trial of Office, the page will encourage them to setup their free trial. This is a cloud service page.
15. **Payment information.** New in Windows 10, version 1803. Only displays if a user opts-in to a free trial of Office from the **Set up Office** screen. This is a cloud service page.
16. **Make Cortana my personal assistant**
17. **Privacy settings.** Users will see up to seven privacy settings on this screen. Not all users will see the same settings.
18. **OEM Registration pages**

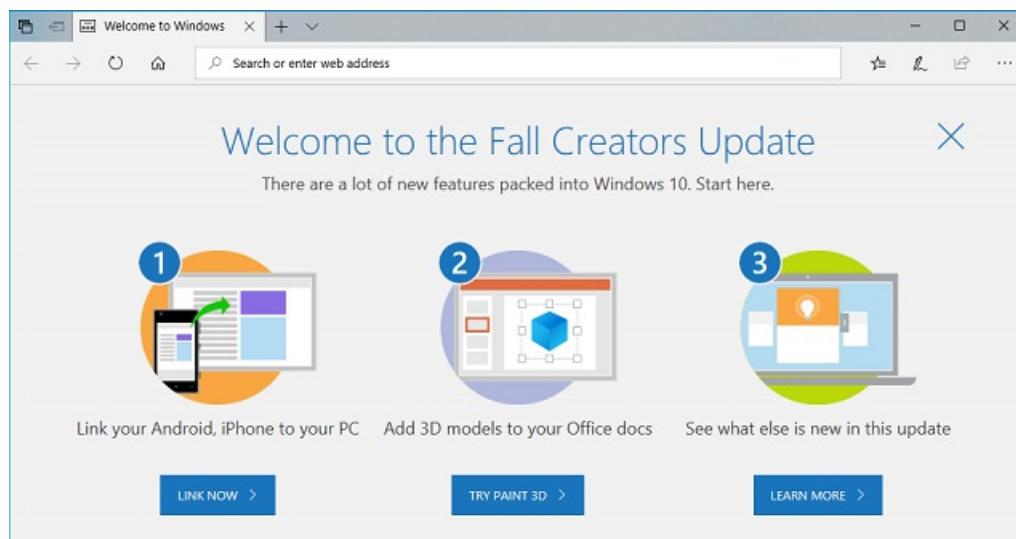
#### **NOTE**

Some pages displayed during OOBE are delivered via cloud service, as opposed to being delivered as part of a Windows release. Cloud service pages can be rolled out to users, or groups of users, at any time. Page content can also be modified or adapted based on user input. Using cloud service for OOBE pages enables Microsoft to offer targeted, relevant content to users quickly, rather than waiting for the next Windows release.

When testing OOBE, keep in mind that you may not see cloud service pages during the flow.

## Windows Welcome

In Windows 10, version 1803, Windows Welcome is displayed to more users than ever as soon as they complete OOBE and reach their desktop. Here's an example Windows Welcome experience:



## In this section

The following topics describe OOBE customization considerations.

TOPIC	DESCRIPTION
<a href="#">OOBE.xml</a>	Use OOBE.xml to organize text and images displayed during OOBE, and to specify settings for customizing the Windows 10 first-run experience. You can use multiple Oobe.xml files for language- and region-specific license terms and settings so that users see appropriate info as soon as they start their PCs. By specifying information in the Oobe.xml file, you help fill in some of the required information so that users are asked to do only the core tasks required to set up their PCs.

Topic	Description
<a href="#">Cortana voice support</a>	Learn how Cortana voice walks the user through the OOBE experience, enabling the user to complete parts of OOBE by responding to spoken prompts.
<a href="#">OOBE screen details</a>	Learn about the <b>Let's connect you to a network</b> , <b>Create security questions</b> , and <b>Payment information</b> screens in OOBE. Although these screens aren't customizable, they are described here to provide insight to the user experience during OOBE.
<a href="#">Windows Updates during OOBE</a>	Learn how both critical and non-critical Windows updates can download during a user's Out of Box Experience.
<a href="#">OEM HID pairing</a>	On PCs that ship with an unpaired wireless mouse and keyboard, you can customize the HID pairing screens shown to the customer during the first-run experience in OOBE. If you include written instructions, you must include those instructions in every language that ships with the PC.
<a href="#">OEM license terms</a>	You can add your OEM license terms to the License Terms screen in the first-run experience of OOBE.
<a href="#">OEM registration pages</a>	You can display OEM registration screens during OOBE to encourage customers to provide you with their information. This enables you to provide them with a more personalized experience and information.
<a href="#">Automate OOBE</a>	Use Unattend settings to hide certain pages that appear in OOBE.

## Related topics

[OOBE Unattend component](#)

# OOBE.xml

10/2/2018 • 2 minutes to read • [Edit Online](#)

Create a file named **Oobe.xml** to organize text and images displayed during OOBE, and to specify settings for customizing the Windows 10 first-run experience. For Windows 10, you can use multiple Oobe.xml files for language- and region-specific license terms and settings so that users see appropriate info as soon as they start their PCs. By specifying information in the Oobe.xml file, you help fill in some of the required information so that users are asked to do only the core tasks required to set up their PCs.

## OOBE.xml settings

You can set the default language, location, and keyboard layout using Oobe.xml. The default values you set in Oobe.xml will be the default values the user sees on the Language, Region, and Keyboard layout selection screens during OOBE. The user can select another value from the list if desired, and their selection will override the Oobe.xml settings.

You can also specify a default timezone for the device using Oobe.xml. If the device has network connectivity during OOBE, Windows will attempt to detect the user's time zone and this will override the value set in Oobe.xml. If the device does not have connectivity, or the user has turned off Location settings in OOBE, Windows will not be able to detect the timezone, and will default to the value you set in Oobe.xml. In this case, the user will see this timezone reflected by their clock once they reach the desktop.

For a list of time zones you can set, see [Default Time Zones](#).

There are a number of other settings available to enable further customization of OOBE. See [Configure Oobe.xml](#) for information about all of the settings available to you.

## Configure OOBE.xml for multi-language and region deployments, and single-language and region deployments

You can create multiple OOBE.xml files for each language and region you intend to deploy in to provide appropriate default values in each location. For more information, see [How OOBE.xml works](#).

## Oobe.xml example

```
<FirstExperience>
<oobe>
<oem>
    <name>Fabrikam</name>
    <eulafilename>eula.rtf</eulafilename>
    <computername>Fabrikam-PC</computername>
    <registration>
        <title>Register your PC</title>
        <subtitle>This page will help Fabrikam know about you.</subtitle>
        <customerinfo>
            <label>Let Fabrikam contact you</label>
            <defaultValue>true</defaultValue>
        </customerinfo>
        <checkbox1>
            <label>Use Contoso Antimalware to help protect your PC</label>
            <defaultValue>true</defaultValue>
        </checkbox1>
        <checkbox2>
            <label>Let Fabrikam send you offers</label>
        </checkbox2>
        <checkbox3>
            <label>Let Fabrikam send you offers</label>
        </checkbox3>
        <link1>
            <label>Learn more about Contoso Antimalware</label>
        </link1>
        <link2>
            <label>Learn more about Fabrikam offers</label>
        </link2>
        <link3>
            <label>Fabrikam privacy statement</label>
        </link3>
        <hideSkip>true</hideSkip>
    </registration>
</oem>
<defaults>
    <language>1033</language>
    <location>244</location>
    <keyboard>0409:00000409</keyboard>
    <timezone>Central Europe Daylight Time</timezone>
    <adjustForDST>true</adjustForDST>
</defaults>
<hidSetup>
    <title>Pair Your Fabrikam MouseKeyboard</title>
    <mouseImagePath>c:\fabrikam\mouse.png</mouseImagePath>
    <mouseErrorImagePath>c:\fabrikam\errormouse.png</mouseErrorImagePath>
    <mouseText>Pair your mouse now.</mouseText>
    <mouseErrorText>Something has gone wrong.</mouseErrorText>
    <keyboardImagePath>c:\fabrikam\keyboard.png</keyboardImagePath>
    <keyboardErrorImagePath>C:\fabrikam\errorkeyboard.png</keyboardErrorImagePath>
    <keyboardText>Now pair the keyboard.</keyboardText>
    <keyboardErrorText>Keyboard pairing did not happen.</keyboardErrorText>
    <keyboardPINImagePath>c:\fabrikam\keyboardpin.png</keyboardPINImagePath>
    <keyboardPINText>Enter the PIN for your keyboard.</keyboardPINText>
</hidSetup>
</oobe>
</FirstExperience>
```

# Cortana voice support

10/8/2018 • 2 minutes to read • [Edit Online](#)

Cortana voice walks the user through the OOBE experience, enabling the user to complete parts of OOBE by responding to spoken prompts. Cortana voice during OOBE is currently available in the following languages: **en-US, es-MX, ja-JP, en-GB, fr-FR, it-IT, de-DE, es-ES, fr-CA, en-CA, en-AU, pt-BR, zh-CN**.

The `language` value you set in OOBE.xml impacts the voice used during OOBE. The OOBE.xml value for `language` must be a language/region decimal ID associated with a Windows language pack. For example, the **English (United States)** language pack has an associated language/region decimal ID of **1033**. For a full list of language/region decimal IDs that you can set in OOBE.xml, see [Available Language Packs for Windows](#).

Cortana voice is enabled after the customer selects a language from the Language selection screen in OOBE. If the language selected by the customer, combined with the `language` in OOBE.xml, is supported by Cortana, Cortana will assist in that language upon entering the Region selection page.

Cortana voice will continue to assist throughout the OOBE process in that same supported language. Even if the user selects a region on the Region page that is not supported by Cortana, or selects a region that would cause Cortana to use a different accent after OOBE, Cortana voice will not change during OOBE.

If the language selected by the customer on the Language page combined with the `language` in OOBE.xml is not one of the supported combinations for Cortana, then the OOBE experience will be silent.

After the user completes OOBE, the voice used in the Cortana app will be based on the Language and Region selected during OOBE. At that point, Cortana will no longer consider the language in OOBE.xml.

Here are a few examples:

LANGUAGE SELECTED BY CUSTOMER (DURING OOBE)	LANGUAGE SET IN OOBE.XML	REGION SELECTED BY CUSTOMER (DURING OOBE)	CORTANA VOICE ASSISTANCE RESULT (DURING OOBE)	CORTANA APP VOICE RESULT (AFTER OOBE)
English	1033 (en-US language pack)	US	en-US	en-US
English	1033 (en-US language pack)	US	en-GB	en-US
Russian	1049 (ru-RU language pack)	US	Silent	Not supported

## Disable Cortana voice support

For testing purposes, you can turn Cortana voice off, but you must enable it again before the device ships. To temporarily turn Cortana voice off, set the following registry key.

`HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\OOBE : DisableVoice = 1 (DWORD)`

### NOTE

This setting should only be disabled for testing purposes. Shipping a device with Cortana voice support disabled is an unsupported configuration.

## "Hey Cortana" feature

The "Hey Cortana" feature enables users to more easily engage Cortana on their Windows 10 device by speaking the phrase "Hey Cortana".

For devices that meet hardware requirements, users have the option of enabling "Hey Cortana" during the OOBE flow, on the screen which asks the user if they'd like to make Cortana their personal assistant. The option is unchecked by default.

After OOBE, users can also enable "Hey Cortana" from **Cortana & Search Settings**. By default, "Hey Cortana" is not enabled.

## Configure Hey Cortana

To optimize battery life, by default, Windows only asks users if they want to enable "Hey Cortana" on desktop devices with a microphone

For Windows 10, version 1709 and later, you can also include this option during OOBE if your device meets the policy requirement of including a [hardware-offloaded key spotter \(HW KWS\)](#).

For devices that meet this requirement, set the registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Speech_OneCore\AudioPolicy : VoiceActivationIsBatteryCertified = 1.
```

To learn more, see [Voice Activation](#).

# OOBE screen details

10/2/2018 • 6 minutes to read • [Edit Online](#)

This topic describes some of the screens users will see as they progress through OOBE. Although the screens described here are not customizable, the information is provided to give insight to the user's experience, and what the user can expect, as they work through OOBE.

The following screens are described below:

- [Cloud service pages](#)
- [Connect users to the network](#)
- [Create security questions for this account](#) (new in Windows 10, version 1803)
- [Set up Office](#)

## Cloud service pages

Some pages displayed during OOBE are delivered via cloud service, as opposed to being delivered as part of a Windows release. Cloud service pages can be rolled out to users, or groups of users, at any time. Page content can also be modified or adapted based on user input. Using cloud service for OOBE pages enables Microsoft to offer targeted, relevant content to users quickly, rather than waiting for the next Windows release.

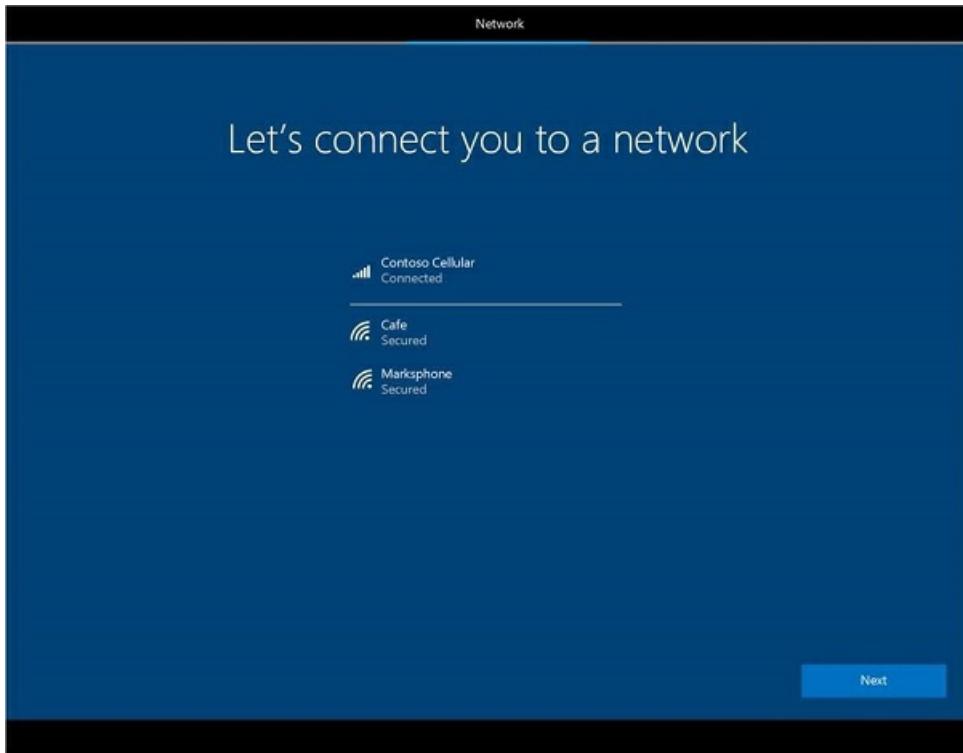
When testing OOBE, keep in mind that you may not see cloud service pages during the flow.

## Connect users to the network

During the OOBE flow, the customer will see the **Let's connect you to a network** screen. This screen appears just prior to the EULA screen during OOBE. **Let's connect you to a network** shows any connection options available to the user, including in-range Wi-Fi and Cellular data networks.

### Connect to Cellular and/or Wi-Fi networks

If the device is LTE-enabled and an active SIM card is inserted, Windows will connect to the Cellular data network automatically. This enables the user to go through OOBE and successfully setup their device if a local Wi-Fi connection is not available. The user will see they are Connected to the Cellular data network when they reach the **Let's connect you to a network** screen in OOBE. Available Wi-Fi connections will also be shown on the screen, and the user can choose to connect to Wi-Fi if desired.



If the device is LTE-enabled, but no SIM card is present, Cellular data will appear as a connection option along with any available Wi-Fi networks. The user must insert a SIM card before they can connect to the Cellular network.

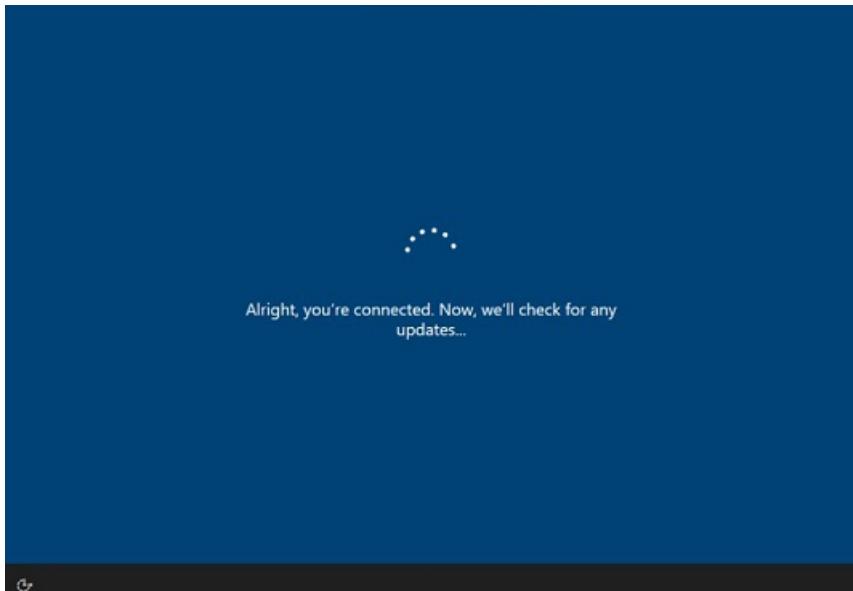
A user can choose to connect to both a Wi-Fi and Cellular network at the same time. In this case, Wi-Fi will be used throughout OOSE and no data traffic is transmitted via the Cellular network (metered connection). Windows will always use the Wi-Fi connection if it is available. Cellular will only be used if the user is out of range of their Wi-Fi network, or chooses to disconnect from Wi-Fi.

Windows has logic in place to protect the user from draining their data during OOSE if they are on a metered connection (either metered Cellular or metered Wi-Fi). For example, if a user is on a metered network, only critical updates (for example, critical driver updates and zero-day patch (ZDP) Windows updates) are allowed on the device.

For more information on the cellular settings for Windows 10 users, see [Cellular settings in Windows 10](#).

### **Download critical updates after connecting**

Immediately after the user connects to a network, critical driver updates, and Windows ZDP updates, will begin downloading to the device. Only critical updates that are required for the device to function, such as security fixes, will download during this time. As such, the user can't opt out of downloading them. Windows will alert the user that the device is checking for, and applying, the updates:



The time required to download the updates depends on the size of the download and the user's network conditions. Their device may restart automatically during the download.

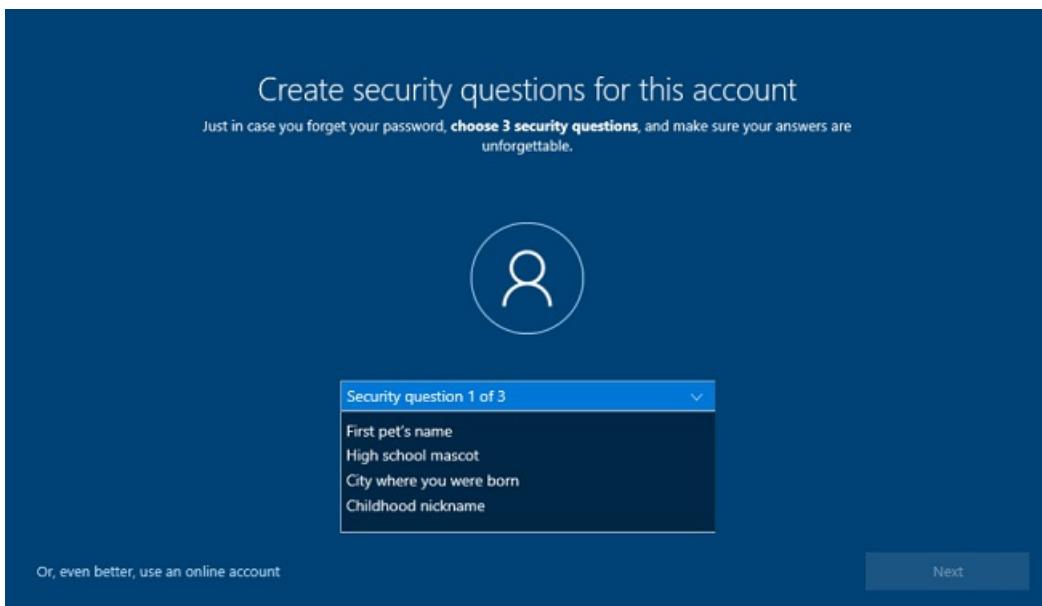
If a newer version of Windows has become available since the device was shipped, the user will be asked if they would like to download this non-critical Windows Update at the end of OOBE. For more information, see [Windows updates during OOBE](#).

## Create security questions for this account

During the OOBE flow, users are prompted to either create or sign in with a local account, or a Microsoft account (MSA). In Windows 10, version 1803, Windows introduces password recovery security questions to accompany the local account registration process in OOBE. If a user is unable to remember the password required to login to the local account, they can instead correctly answer their 3 security questions, and Windows will allow them to reset the password and login to the device.

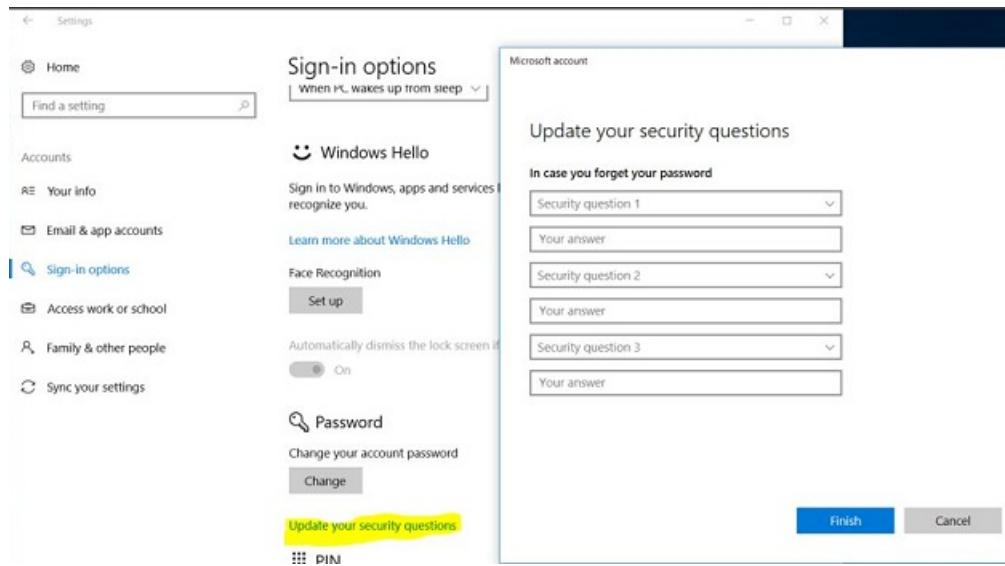
This is a change from previous versions of Windows, where a user was only able to create a password hint to accompany their local account. Previously, if a user couldn't remember their local account password based on the hint, they were required to contact Microsoft support for a device reset.

The **Create security questions for this account** screen will appear after the user creates a local account (name and password) for the device during OOBE.



The list of security questions the user can choose from is generated by Microsoft.

Users can create and update the security questions associated with their local account after OOBE, from the **Settings** app.



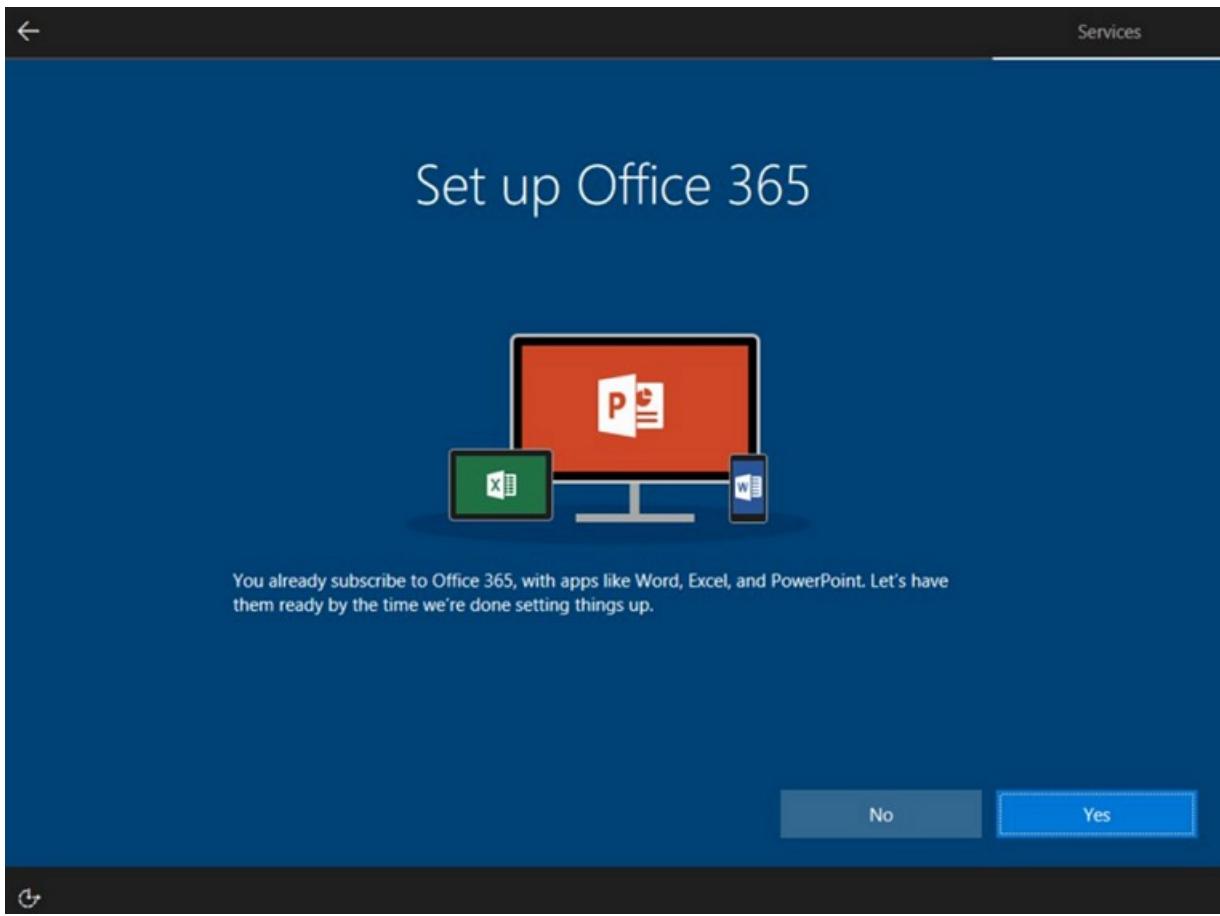
The option to create security questions from the **Settings** app is also available to users who upgrade to Windows 10, version 1803 from a previous version of Windows, and to any new local account created for a device running Windows 10, version 1803.

## Set up Office

Users will see the **Set up Office** screens in OOBE if they are connected to a network, and have provided their Microsoft account (MSA) information. Content on the page will vary depending on the user's account type, and the version of Office pre-installed to the device. The **Set up Office** screens, including the **payment information** screen, are cloud service pages.

### Office 365 subscribers

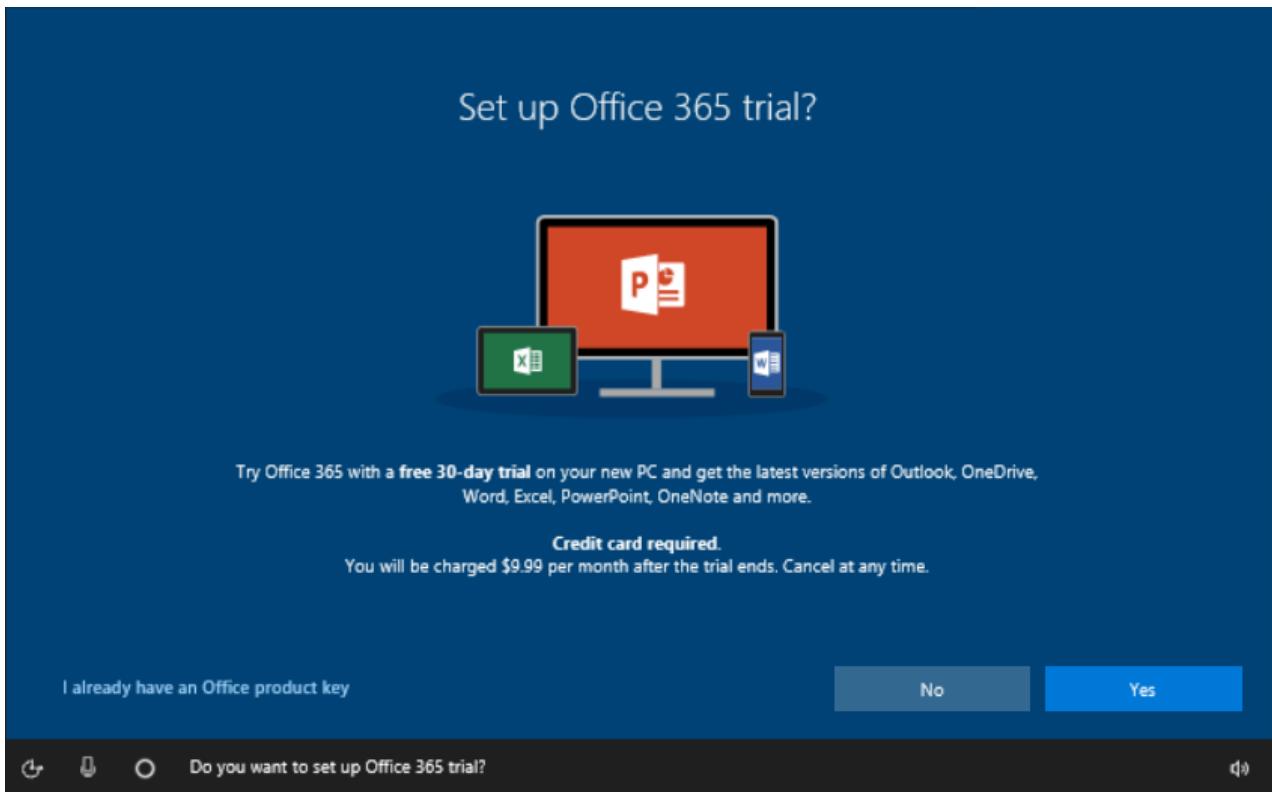
If Office 365 (Office Desktop Bridge) is pre-installed to the device, and the user's MSA is already associated with an Office 365 subscription, they will see the following **Set up Office 365** screen:



The screen reminds the user of their existing subscription, and asks if they would like to have their Office apps ready by the time OOME is complete.

#### Office 365 free trial

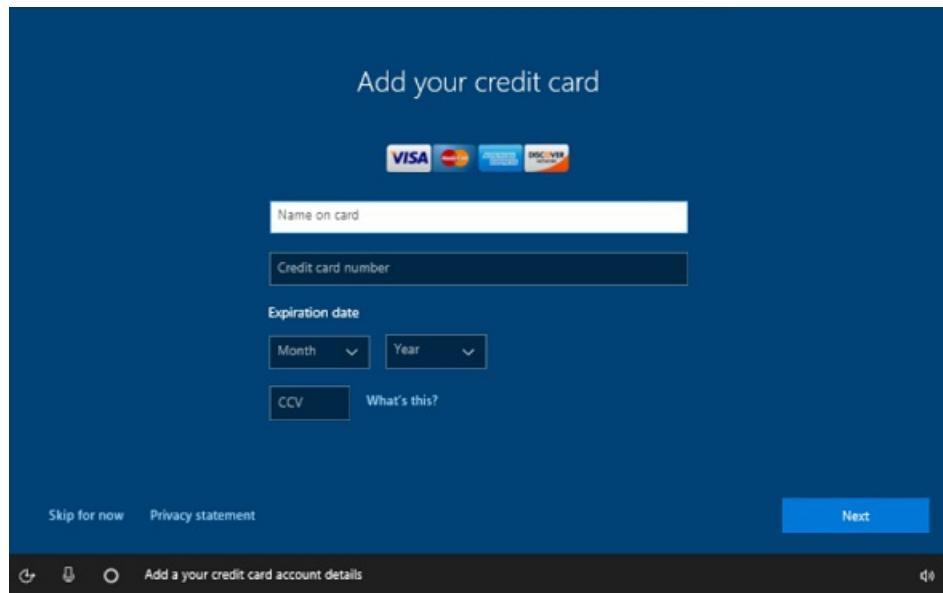
If Office 365 (Office Desktop Bridge) is pre-installed to the device, and the user's MSA is eligible for a free trial, they will see the following **Set up Office 365 trial** screen:



The user can choose to begin the trial by clicking **Yes**, or can opt-out of the trial by clicking **No**.

#### Add credit card information

In Windows 10, version 1803, if a user opts-in to the free trial, they are prompted to enter the credit card to charge when the free trial expires.



If a user is eligible for a free trial of Office 365, and they already have a credit card on file for their Microsoft account, they will not be prompted to **Add your credit card** during OOBE. Instead, they will be asked to confirm that the credit card on file should be charged when the free trial expires.

Collecting this payment information during OOBE enables customers to seamlessly auto-renew Office 365 after the free trial, with no disruption to their service. The credit card will be saved to the user's MSA, so it can be used for future purchases.

A credit card is now required to start a free trial of Office 365. If the user does not provide their payment information during OOBE, they will not be able to start a free trial at that time. The customer can start a free trial of Office 365 later but will still be required to enter their payment information.

#### Office 2016

If Office 2016 (Activation For Office (AFO) Perpetual) is pre-installed to the device, users will see the following **Set up Office** screen:

# Set up Office



You already have Office as part of the machine with apps like Word, Excel and PowerPoint. Let's have them ready by the time we are done setting things up

No

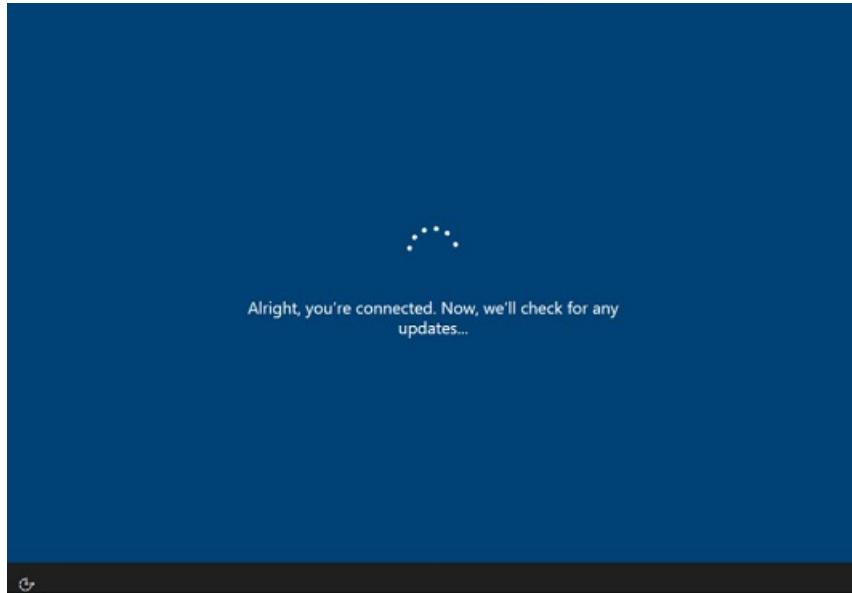
Yes

The screen informs the user that Office 2016 is included with their device, and asks if they would like to have their Office apps ready by the time OOBE is complete.

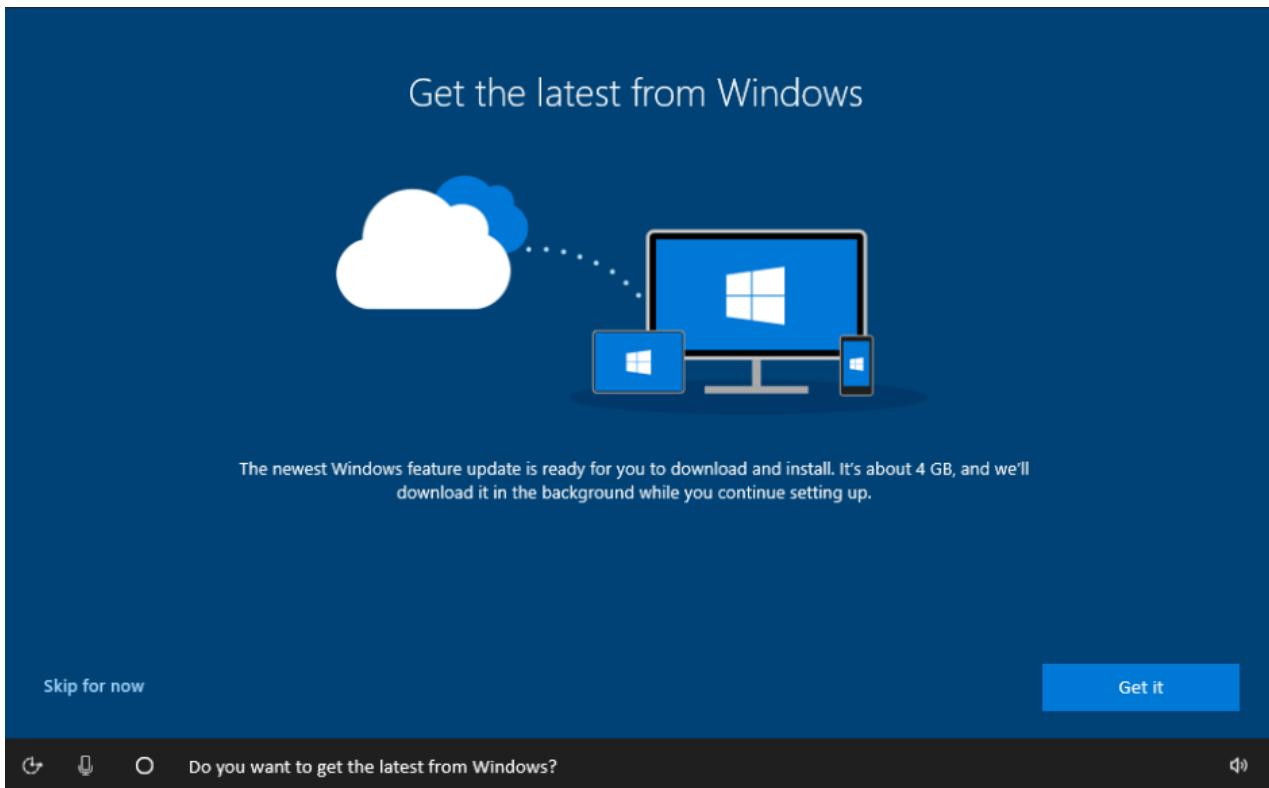
# Windows updates during OOBE

10/2/2018 • 2 minutes to read • [Edit Online](#)

Critical driver updates, and critical Windows zero-day patch (ZDP) updates, will begin downloading automatically during OOBE after the user has [connected to a network](#). The user can't opt-out of these critical updates as they are required for the device to operate properly. Windows will alert the user that the device is checking for, and applying, the updates:



A user can also opt-in to **Get the latest from Windows** during OOBE, if a newer version of Windows is available than the version that shipped with the device. Version updates are considered non-critical, as the device will still continue to perform well after OOBE if the user does not download the update. In Windows 10, version 1803, the **Get the latest from Windows** screen is displayed right after the **Let's connect you to a network** screen in OOBE. This is a change from previous versions of Windows, where this screen had a different title and was displayed at the end of OOBE.



#### NOTE

Users will only see this screen in OOBE if there is a newer version of Windows available than the version that shipped with the device. For example, the screen above will be displayed on devices shipped with Windows, version 1803, but only after the next version of Windows is available.

This screen informs the user of the size of the update. The size of the update, and the user's network conditions, will determine the download time.

The user has the option to click **Get it** or **Skip for now**. In either case, the user's selection will not disrupt their progression through OOBE. Clicking either **Get it** or **Skip for now** will cause the user to move to the next screen in OOBE.

If the user clicks **Get it**, the Windows update will begin downloading as soon as the user has completed OOBE and reached their desktop. It will not begin downloading during OOBE. The user will see a toast message letting them know that the download is taking place, and they will be prompted to restart the device when Windows is ready to install the update. They can continue to use their device while the latest version of Windows is downloading, although performance may be impacted.

If the user selects **Skip for now**, the Windows update will not download after the user has completed OOBE and reached their desktop. The user can choose to download the update at a time of their choosing from the **Settings** app in Windows.

# OEM HID pairing

10/2/2018 • 4 minutes to read • [Edit Online](#)

You can provide clear and precise Human Interface Device (HID) pairing instructions within OOBE to enable customers who buy new PCs running Windows 10 with an unpaired wireless mouse and keyboard to finish their PC setup. For this feature to work, the mouse and/or keyboard must be included with the PC, and the PC must not have any other mice or keyboards attached or connected to it. For example, laptops are not qualified for this feature.

The following conditions should be met to correctly display HID pairing screens during OOBE:

- The PC must have Bluetooth capability and Bluetooth must be turned on.
- The Bluetooth radio must be certified for Windows 10.
- For the keyboard pairing page to appear, you must ensure that no wired keyboard is connected to the PC.
- For the mouse pairing page to appear, you must ensure that no wired mouse is connected to the PC.
- Oobe.xml settings in the `<hidsetup>` section should be provided for the corresponding pairing pages.

We recommend that OEMs include Bluetooth radios with HEM to provide a working end-to-end scenario, because there is no Bluetooth support in the BIOS before Windows loads. The radio looks like a USB mouse and keyboard to the PC and takes over the Bluetooth communication to the mouse and keyboard. This lets the devices work outside of Windows and allows customers to use their paired Bluetooth mouse and keyboard during BIOS.

## IMPORTANT

The Oobe.xml file that has HID pairing instructions must be used only for PCs that use the OOBE HID pairing feature. For PCs that don't use the OOBE HID pairing feature, a different Oobe.xml file that doesn't contain the HID pairing instructions must be used. Otherwise, there's a risk that users might go through the HID pairing experience even if they don't need to or can't use this feature.

## Configure Oobe.xml

On PCs that ship with an unpaired wireless mouse and keyboard, the HID pairing screens are shown to the customer during the first experience, which is before language selection or any other screen that requires user input in OOBE. You can also choose to include written instructions, however, if you do this you must include those instructions in every language that ships with the PC.

To provide a thorough, reliable, and satisfactory HID pairing experience, OEMs who ship these systems must include the following Oobe.xml settings:

Oobe.xml setting	Description
<code>&lt;mouseImagePath&gt;</code>	The path to a mouse pairing instruction image. The three steps customers typically perform are inserting batteries into the mouse, turning on the power, and turning on Bluetooth.
<code>&lt;mouseErrorImagePath&gt;</code>	The path to a mouse pairing error image. If the customer can't pair the mouse in three tries, this error screen appears.

OOBE.XML SETTING	DESCRIPTION
<keyboardImagePath>	The path to a keyboard pairing instruction image. The first three steps customers typically perform are inserting batteries into the keyboard, turning on power, and turning on Bluetooth. You can include these steps in the first image. Usually the second set of steps customers need to perform are entering a password or code and pressing Enter.
<keyboardPINImagePath>	The path to a keyboard pairing instruction image.
<keyboardErrorImagePath>	The path to a keyboard pairing error image. If the customer can't pair the keyboard in five tries, this error screen appears. This should tell the customer to connect a wired keyboard.
<mouseText>	Help text that displays at the bottom of the page.
<mouseErrorText>	Error that displays to users along with mouse pairing error image.
<keyboardErrorText>	Error that displays to users along with keyboard pairing error image.
<keyboardText>	Specifies the text to prompt the user to pair the keyboard.
<keyboardPINText>	Specifies the prompt text for the user to enter a pin for the keyboard.

Any text in the Oobe.xml file or files — for example, any text in the `<mouseText>` setting — is the text read by the Narrator, so make sure the text is clear, concise, and easy to understand. Cortana shares duties with Narrator, so that Cortana speaks the UI display text and Narrator speaks the instructional text.

For more information on these settings, see [Oobe.xml settings](#)

## Guidelines for images

We recommend that you use photorealistic images of the HID devices that ship with the system. This will help customers realize that the instructions you provide apply to the hardware they have just purchased. The illustration examples we've provided are generic. We do not use photorealistic images so our documentation does not appear to apply to any one device or hardware partner.

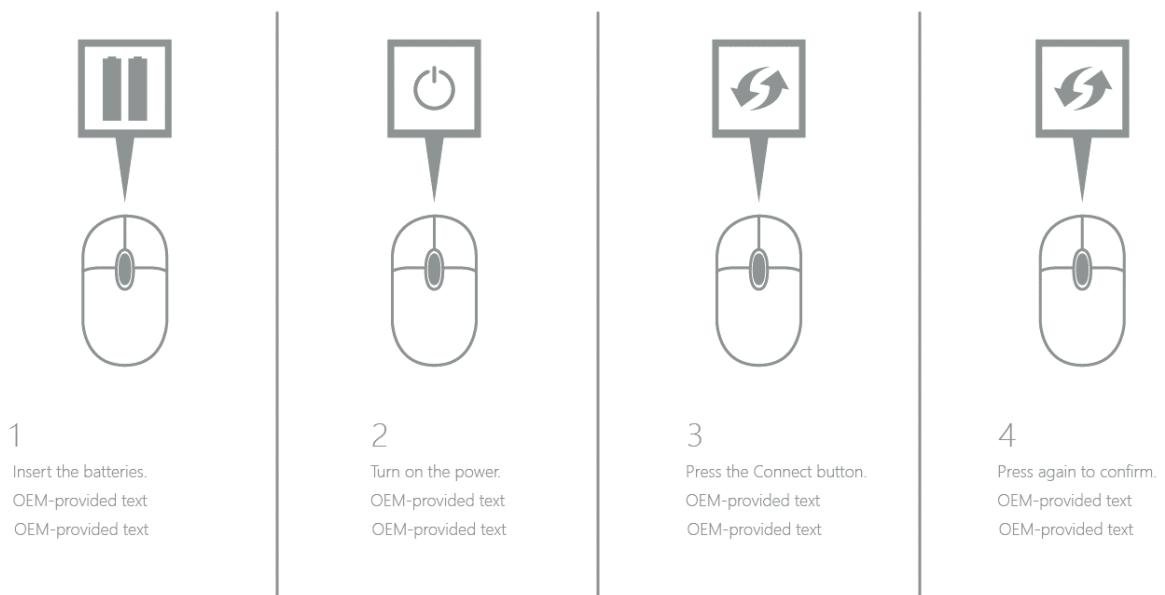
Generic images decrease confidence and increase confusion for customers, who want images on the screen to match the devices they're trying to use. Also, include visual instructions for the actions customers must take to pair their new hardware. For example, if the first step is to insert batteries into the device, include an image of batteries near the device.

### NOTE

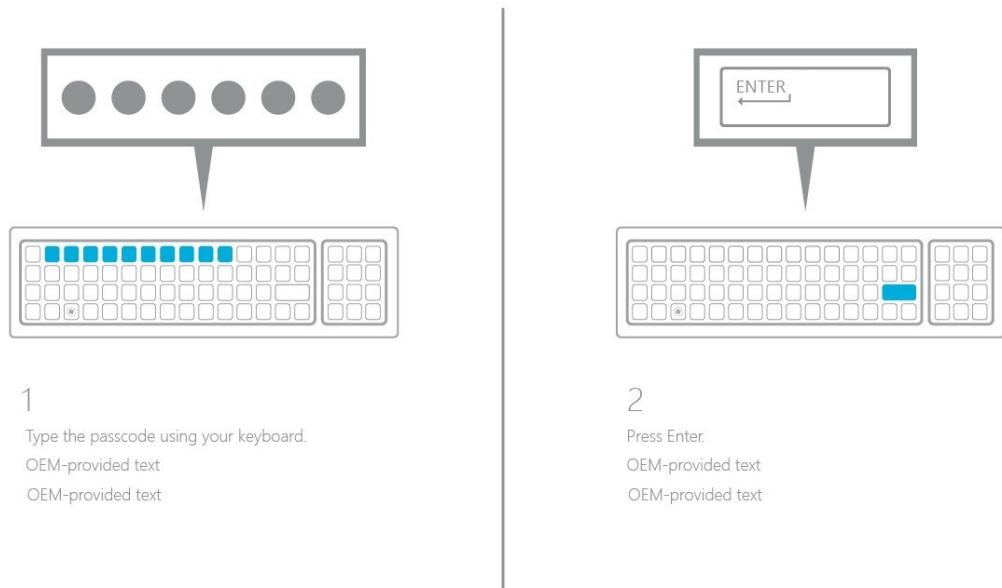
Images must not be larger than 630 x 372 pixels. Images are scaled to fit in portrait mode or on small form factors.

These illustrations are examples of how HID pairing instructions might look:

### Example 1: Image for mouse pairing



### Example 2: Image for keyboard pairing



## XML example

```

<hidSetup>
    <mouseImagePath>c:\fabrikam\MouseFirstInstruction.png</mouseImagePath>
    <mouseText>Set up your Fabrikam mouse. Insert batteries, turn on, and press the Bluetooth button.</mouseText>
    <mouseErrorImagePath>c:\fabrikam\MouseError.png</mouseErrorImagePath>
    <mouseErrorText>An error has occurred. Please contact Fabrikam.</mouseErrorText>
    <keyboardImagePath>c:\fabrikam\KeyboardFirstInstruction.png</keyboardImagePath>
    <keyboardText>Set up your Fabrikam keyboard. Insert batteries, turn on, and press the Bluetooth button.</keyboardText>
    <keyboardPINImagePath>c:\fabrikam\KeyboardSecondInstruction.png</keyboardPINImagePath>
    <keyboardPINText>Enter PIN and press the Enter key.</keyboardPINText>
    <keyboardErrorImagePath>c:\fabrikam\KeyboardError.png</keyboardErrorImagePath>
    <keyboardErrorText>An error has occurred. Please contact Fabrikam.</keyboardErrorText>
</hidSetup>

```

# OEM license terms

10/2/2018 • 11 minutes to read • [Edit Online](#)

You can add your OEM license terms to the License Terms screen in the first-run experience. These terms will appear next to the Windows License Terms.

To add your license terms:

1. Create a version of your End User License Agreement (EULA) in RTF (.rtf).
2. Create a version of your EULA in HTML (.html). Files with an .htm extension are ignored. All HTML files in OOBE must use UTF-8 encoding. See [HTML EULA example](#) for an example.
3. The names of your EULA files should be identical, except for the extension (.rtf and .html).
4. Place both versions of your EULA in the `Windows\System32\Oobe\Info` directory, or in subdirectories that you create per the country or region and languages of the image you're shipping. For more information on how to configure subdirectories for multi-language and region deployments, see [How OOBExml works](#).
5. In your Oobe.xml file, set the `<eulafilename>` value to the absolute path of your RTF EULA. You do not need to include the path to the HTML EULA in Oobe.xml. The system will correctly handle both files as long as they have the same name and are stored in the same location. See [Oobe.xml Settings](#) for more information on this setting.

## IMPORTANT

Do not add links to your EULA, as the user should not navigate away from the license terms.

You must include a version of the EULA in each language that you preinstall onto the PC. If you don't include terms for a specific language, an English (EN) version of the license terms displays. The terms must be specific to each language, but they don't need to be specific to each country or region that uses the language. Although the acceptance of the terms isn't recorded, customers can't proceed unless they accept them.

## HTML EULA example

```
<!DOCTYPE html>
<html dir="ltr">
<head>
    <meta charset="utf-8"/>
    <meta name="viewport" content="width=device-width">
    <title></title>
    <style type="text/css">
        [dir='rtl'] dir {
            padding-right: 12px; }

        [dir='ltr'] dir {
            padding-left: 12px; }

        [dir='ltr'] [align=right] {
            text-align: right; }

        [dir='ltr'] [align=left] {
            text-align: left; }

        [dir='rtl'] [align=right] {
            text-align: left; }

        [dir='rtl'] [align=left] {
```

```
        text-align: right; }

[dir='rtl'] body {
    padding-left: 12px;
}

[dir='ltr'] body {
    padding-right:12px;
}

body {
    -ms-overflow-style:scrollbar;
    background:#004275;
    color:#FFF;
    font-family:"Segoe UI", Selawik, Tahoma, Verdana, Arial, sans-serif;
    font-size:.9375rem;
    font-weight:400;
    line-height:1.25rem;
    margin:0;
    max-width:100%;
    overflow:auto;
    padding-bottom:0;
    padding-top:0;
}

body b * {
    font-weight:700;
}

html {
    font-size:100%;
}

p {
    font-size:.9375rem;
    font-weight:400;
    line-height:1.25rem;
    max-width:100%;
    padding-bottom:.0141875rem;
    padding-top:.0141875rem;
}
</style>
</head>
<body>
<B><FONT SIZE=3><P ALIGN="RIGHT">Last updated: </P>
<P>PRE-RELEASE SOFTWARE LICENSE</P>
<P></P>
<P></P>
<P></P>
</B>
<P>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</P>
<P>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</P>
<P>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</P>
<B><P>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</P>
<DIR>
<DIR>

<P>Overview.</P>
<DIR>

<P>Applicability
```





```
commodo consequat.</P>
<P><B>Malware protection.</B> Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</P></DIR>

<B><P>DISCLAIMER OF WARRANTY.</B>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. </P>
<B><P>Limitation on and Exclusion of Remedies and Damages.</B>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. <B>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</B> Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</P>
<P>Entire Agreement.</B> Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</P>
<DIR>
<DIR>
<DIR>

<P>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</P>
<P>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</P>
<P>ALorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</P>
<P></P></DIR>
</DIR>
</DIR>
</DIR>
</FONT>
</body>
</html>
```

# OEM registration pages

10/2/2018 • 2 minutes to read • [Edit Online](#)

You can customize OEM registration pages to gather customer information, and introduce offers, during OOBE. If you choose to implement the optional registration pages, we recommend that you use them to provide information and opportunities that benefit your customers. The Windows 10 OOBE is designed to maximize customer engagement by creating pages that focus on one thing at a time. As a result, OEM registration fields are divided between two separate pages.

Here is an example of the two OEM registration pages:

The image contains two screenshots of the Windows 10 OEM registration process. The top screenshot shows the first registration page titled 'Register your PC' with fields for First name, Last name, Region, and Email. The bottom screenshot shows the second registration page with checkboxes for letting the company contact you, using Contoso Antimalware, and receiving offers, along with links to learn more about these features.

The OEM registration pages work with a Microsoft Account (MSA) to help customers enter in their information only once during OOBE. Microsoft prompts customers to sign up for an MSA or sign into an existing MSA during OOBE. When a customer does this, their first name, last name, and email address for the MSA, if provided, will be pre-filled in on registration page one. The customer can change their information before clicking **Next** if desired.

If the customer has not used an MSA, the fields on the OEM registration pages will be empty, and the customer

can fill them in if and as desired.

The OEM registration pages are the last screens in the OOBE flow, after the user goes through all other steps in OOBE.

The customer information submitted through the registration pages will be stored in the `%systemroot%\System32\Oobe\Info\` folder, and will be encrypted using a public key that you place into the Windows image. Collect the encrypted data using a Microsoft Store app designated as your OEM App, or write a service that does this, and upload the data to your server. Decrypt the data using the corresponding private key once it's on your server.

## In this section

The following topics describe how to add your registration pages to OOBE.

TOPIC	DESCRIPTION
<a href="#">Design your registration pages</a>	Guidance on customizing the registration page fields and HTML flyout pages.
<a href="#">Configure Oobe.xml</a>	The elements of Oobe.xml are used to customize your registration pages. Create a custom Oobe.xml file or files as determined by the languages and regions where you ship your company's PCs. You can use multiple Oobe.xml files for language- and region-specific terms and settings so users see the correct language as soon as they start their PCs.
<a href="#">Protect and collect user data</a>	To protect customer privacy, Windows encrypts the customer data that's created via the Registration pages using a public key that you generate and store in the Windows image. Create an OEM App or write a service that collects the encrypted data and uploads it to your server using SSL. You can then decrypt the data using the corresponding private key.

# Design your registration pages

10/2/2018 • 8 minutes to read • [Edit Online](#)

The OEM registration pages present many customization opportunities. This topic describes all elements on each of the two OEM registration pages, indicating the customization options for each element. This topic also provides style guidance and code samples you can use to design your HTML flyout pages.

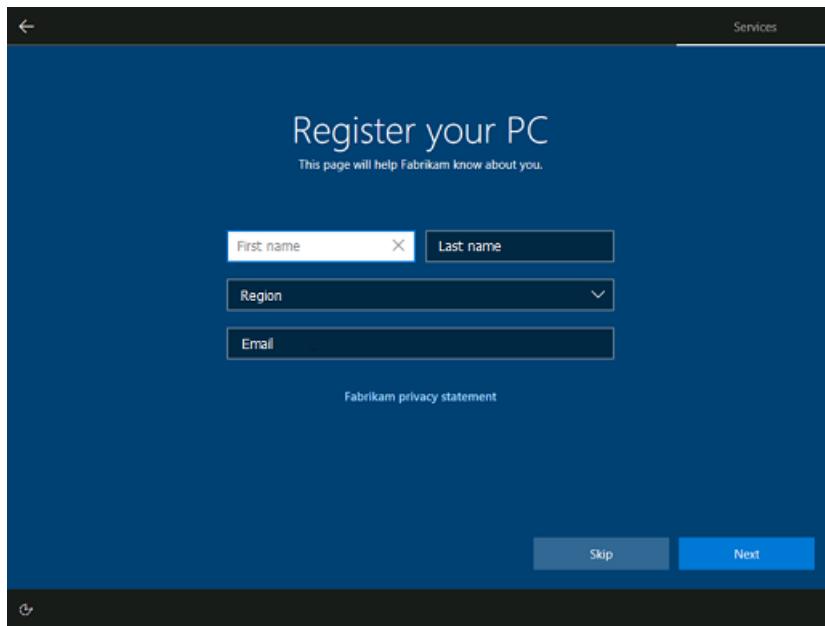
The layout of both OEM registration pages are locked, so the page elements themselves can't be rearranged.

## NOTE

A minimum amount of information is required for the registration pages to display. You must provide a page title, a page subtitle, the `customerinfo` element, at least one additional checkbox **or** one link, and a public key for public/private key encryption.

## OEM registration page one

The first OEM registration page includes the elements below, some of which you can customize.



- **Page title.** Create a title that makes sense for your use of the page. This title also appears on registration page two.
- **Page subtitle.** Add a subtitle to help customers understand the tasks on the page or in some other way guide them to complete the form. This subtitle also appears on registration page two. The page title and subtitle can be customized using the `registration` element of [Oobe.xml](#).
- **Customer information fields.** These fields are not customizable. Customer information consists of four input fields: First Name, Last Name, Region, and Email. If the Email field is filled in, it will be validated as well-formed prior to allowing the customer to proceed. The Country/Region input field is a drop-down list. The associated value of each country/region is its associated two-letter country/region code based on [ISO 3166-1 Alpha-2](#).
- **One link.** Customize the title, and path to, an HTML file using the `link1` element of [Oobe.xml](#). When using this link to surface a privacy policy, ensure the policy is current.
- **Skip button.** The Skip button is visible by default, but you can configure the `hideSkip` element of [Oobe.xml](#) to hide it. No registration data of any kind is provided if the customer chooses **Skip**. The button text is not

customizable.

- **Next button.** The Next button moves the customer forward in OOBE. This button is not customizable.

### Pre-populated customer information

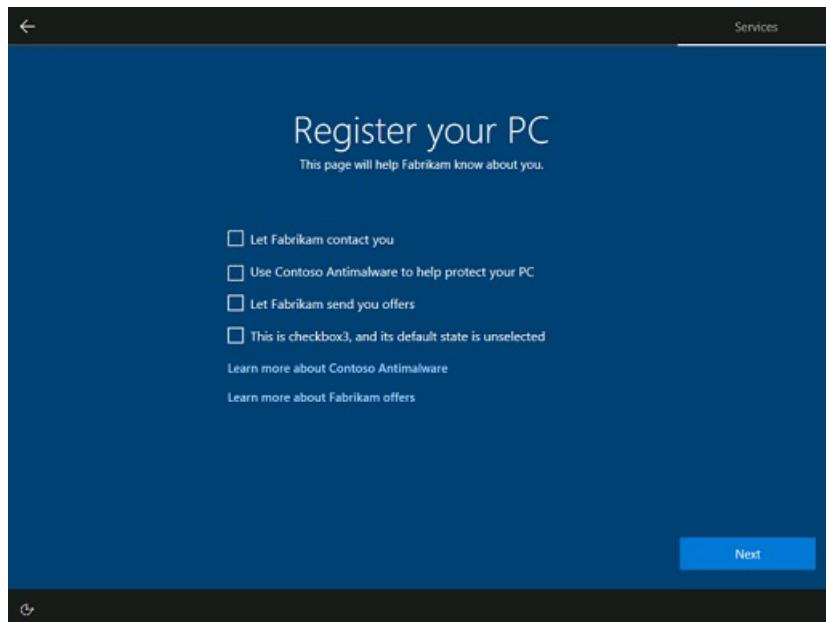
When a user signs in to or signs up for an MSA in OOBE, they provide some of the customer information requested on the OEM Registration pages. To streamline the setup process for users, Windows pre-populates some of the customer information fields on OEM registration page one, if the customer used an MSA earlier in OOBE.

Depending on the SKU a user may choose to setup different account choices which will impact whether the account information is pre-filled.

ACCOUNT PATH	OEM PAGE PRE-FILLED
Microsoft account sign up	First name, last name, email
Microsoft account sign in	First name, last name, email
Azure AD account sign in	Nothing pre-filled
Local account creation	Nothing pre-filled

## OEM registration page two

The second OEM registration page includes the elements below, some of which you can customize.



- **Page title.** Create a title that makes sense for your use of the page. This title also appears on registration page one.
- **Page subtitle.** Add a subtitle to help customers understand the tasks on the page or in some other way guide them to complete the form. This subtitle also appears on registration page one. The page title and subtitle can be customized using the `registration` element of [Oobe.xml](#).
- **Four checkboxes.** Up to four checkboxes with labels can be displayed on registration page two. You can set the descriptive labels for the checkboxes, and their default states, using the `customerinfo`, `checkbox1`, `checkbox2`, and `checkbox3` elements of [Oobe.xml](#).
- **Two links.** Up to two links can be displayed beneath the checkboxes. You can specify the link labels and file paths using the `link2` and `link3` elements of [Oobe.xml](#). Any text you associate with these links must be in

HTML files stored locally in the `%systemroot%\system32\Oobe\Info` directory.

- **Next button.** The Next button moves the customer forward in OOBE. This button is not customizable.

#### NOTE

You can't skip showing a link on registration page one by providing only `link2` and `link3` elements in Oobe.xml. A missing `link1` will cause the `link2` element to appear on the first registration page instead of the second.

## Design HTML files for your links

When a customer clicks any link you've added to the registration pages, this opens an HTML file stored in the `%systemroot%\system32\Oobe\Info` folder on the device. Microsoft provides a full HTML sample below that defines the background color, font color, font sizing, font weight, padding, margins, and headers (among other elements) for your HTML files. We strongly encourage you to use this sample with little to no alteration of the design elements.

Windows OOBE has a dark blue background with light text. End User License Agreement (EULA) content uses a dark blue background and light text. Fly-out content uses a dark background with light text. To align with the design of Windows OOBE, and to create a consistent user experience, use the markup and style conventions laid out in the [HTML example below](#) when creating your HTML files.

#### NOTE

Inline CSS styling is required so that the iFrame host elements render correctly in the registration pages.

### Colors

Text and background colors are defined in the [CSS code example](#).

- Background color: #2b2b2b
- Font color: #FFF

Please use these colors to ensure a consistent user experience throughout OOBE.

### Font

The standard font used throughout OOBE is Segoe UI. Please use the Segoe UI Webfont for your HTML documents to ensure the font matches the rest of OOBE.

### Sizes and spacing

Use two different styles for headers and body content.

- Headers: should be rendered using the `<h4>` tag.
- Body text: should be rendered using the `<p>` tag.
- Bold text: should be rendered using the `<b>` tag.
- Hierarchy of information: indented sections or groups of bulleted items can be displayed with the `<DIR>` tag, required for EULA content template, optional for Flyouts.

We require that the files for the in-place links are HTML. These files are rendered in a flyout. Documents in the flyout are sandboxed, such that links to external and online resources will not function.

## IMPORTANT

The following tags are prohibited and should not be included in your files:

- `<script>`
- `<iframe>`
- `<input>`
- `<img>`
- `<a>`

## CSS code example

Please use the following inline CSS in the head of your HTML documents.

```
<style type="text/css">
    [dir='rtl'] dir {
        padding: 0 12px;
    }

    [dir='ltr'] dir {
        padding: 0 12px;
    }

    [dir='ltr'] [align=right] {
        text-align: right;
    }

    [dir='ltr'] [align=left] {
        text-align: left;
    }

    [dir='rtl'] [align=right] {
        text-align: left;
    }

    [dir='rtl'] [align=left] {
        text-align: right;
    }

    [dir='rtl'] body {
        padding: 0 12px;
    }

    [dir='ltr'] body {
        padding: 0 12px;
    }

    body {
        -ms-overflow-style: scrollbar;
        background: #2b2b2b;
        color: #FFF;
        font-family: "Segoe UI", "Segoe UI Webfont", "Ebrima", "Nirmala UI", "Gadugi", "Segoe Xbox
Symbol", "Segoe UI Symbol", "Meiryo UI", "Khmer UI", "Tunga", "Lao UI", "Raavi", "Iskoola Pota", "Latha",
"Leelawadee", "Microsoft YaHei UI", "Microsoft JhengHei UI", "Malgun Gothic", "Estrangelo Edessa", "Microsoft
Himalaya", "Microsoft New Tai Lue", "Microsoft PhagsPa", "Microsoft Tai Le", "Microsoft Yi Baiti", "Mongolian
Baiti", "MV Boli", "Myanmar Text", "Cambria Math", Selawik, Tahoma, Verdana, Arial, sans-serif;
        font-size: .9375rem;
        font-weight: 400;
        line-height: 1.25rem;
        margin: 0;
        max-width: 100%;
        overflow: auto;
        padding-bottom: 0;
    }
</style>
```

```

        padding-top: 0;
    }

    body b * {
        font-weight: 700;
    }

    html {
        font-size: 100%;
    }

    p {
        font-size: .9375rem;
        font-weight: 400;
        line-height: 1.25rem;
        max-width: 100%;
        padding-bottom: .0141875rem;
        padding-top: .0141875rem;
    }

    h4 {
        font-size: 1.25rem;
        font-weight: 400;
        line-height: 100%;
        max-width: 100%;
        padding-top: 12px;
        margin: 0;
    }

```

</style>

## Full HTML example

Here is a full example of an HTML flyout for OEM registration pages. Please use this sample as a baseline for your HTML flyout pages, with little to no alteration of the design elements.

```

> <!DOCTYPE html>
<html dir="ltr">
<head>
    <meta charset="utf-8"/>
    <meta name="viewport" content="width=device-width">
    <title></title>
    <style type="text/css">
        [dir='rtl'] dir {
            padding: 0 12px;
        }

        [dir='ltr'] dir {
            padding: 0 12px;
        }

        [dir='ltr'] [align=right] {
            text-align: right;
        }

        [dir='ltr'] [align=left] {
            text-align: left;
        }

        [dir='rtl'] [align=right] {
            text-align: left;
        }

        [dir='rtl'] [align=left] {
            text-align: right;
        }

```

```

[dir='rtl'] body {
    padding: 0 12px;
}

[dir='ltr'] body {
    padding: 0 12px;
}

body {
    -ms-overflow-style: scrollbar;
    background: #2b2b2b;
    color: #FFF;
    font-family: "Segoe UI", "Segoe UI Webfont", "Ebrima", "Nirmala UI", "Gadugi", "Segoe Xbox
Symbol", "Segoe UI Symbol", "Meiryo UI", "Khmer UI", "Tunga", "Lao UI", "Raavi", "Iskoola Pota", "Latha",
"Leelawadee", "Microsoft YaHei UI", "Microsoft JhengHei UI", "Malgun Gothic", "Estrangelo Edessa", "Microsoft
Himalaya", "Microsoft New Tai Lue", "Microsoft PhagsPa", "Microsoft Tai Le", "Microsoft Yi Baiti", "Mongolian
Baiti", "MV Boli", "Myanmar Text", "Cambria Math", Selawik, Tahoma, Verdana, Arial, sans-serif;
    font-size: .9375rem;
    font-weight: 400;
    line-height: 1.25rem;
    margin: 0;
    max-width: 100%;
    overflow: auto;
    padding-bottom: 0;
    padding-top: 0;
}

body b * {
    font-weight: 700;
}

html {
    font-size: 100%;
}

p {
    font-size: .9375rem;
    font-weight: 400;
    line-height: 1.25rem;
    max-width: 100%;
    padding-bottom: .0141875rem;
    padding-top: .0141875rem;
}

h4 {
    font-size: 1.25rem;
    font-weight: 400;
    line-height: 100%;
    max-width: 100%;
    padding-top: 12px;
    margin: 0;
}
</style>
</head>
<body>
<H4>Learn more about the sample</H4>
<P>Quisque efficitur lorem nec mauris semper consequat. Aliquam sollicitudin rhoncus sollicitudin. Integer
ligula mauris, euismod ac lacus et, cursus pulvinar mauris. Aliquam sollicitudin blandit vehicula. Morbi ac
arcu vitae mi placerat facilisis eu sed enim. Ut ornare aliquet tincidunt. Maecenas posuere et nisi in tempor.
</P>
<B><P>Donec malesuada bibendum nibh, in semper nunc efficitur sit amet. Vestibulum vehicula hendrerit elit et
congue.</P>
<DIR>
<DIR>

<P>1.&#9;Pellentesque mollis cursus ultrices.</P>
<DIR>

<P>a.&#9;Vivamus ut suscipit arcu.

```

```
</B> Donec viverra tortor lacus, eu aliquam dolor auctor quis. Praesent eget tincidunt metus, non pellentesque  
metus. </P>  
<B><P>b.&#9;Nulla tincidunt urna et tortor gravida, id dictum ligula lacinia.</B> Vivamus libero mauris,  
fermentum et pharetra id, ultricies quis urna.</P>  
<DIR>  
<DIR>  
  
<P>(i)&#9;Suspendisse porta vestibulum risus, et molestie est egestas ut.</P>  
<P>(ii)&#9; Nullam feugiat, odio vel convallis fringilla, libero nibh volutpat metus, a ultrices justo est  
id nisl.</P>  
<P>(iii)&#9;Nunc vulputate turpis at eleifend malesuada.</P>  
<P>(iv)&#9;Cras maximus mi porta arcu vehicula elementum.</P></DIR>  
</DIR>  
</DIR>  
  
<B><P>2.&#9;Nullam ullamcorper placerat finibus.</B> Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
Donec vitae tincidunt quam, viverra vehicula urna. Sed sit amet volutpat ex, id egestas odio.  
Aliquam at urna mollis, commodo ex sit amet, auctor erat. Proin elit neque, pretium ut lorem eget, cursus  
condimentum ante. Quisque placerat tempor nunc, a pulvinar augue interdum sit amet. Sed eget sem quis tellus  
rutrum rhoncus. Suspendisse potenti. Vestibulum sem ipsum, volutpat ac condimentum ut, porttitor ac nulla.  
Quisque rhoncus sapien eu dolor posuere, ac auctor mi dapibus. Aenean egestas mauris sed tellus dapibus, sed  
sagittis velit volutpat:</P>  
<DIR>  
<DIR>  
<DIR>  
  
<P>·&#9;Sed mattis varius libero.</P>  
<P>·&#9;Maecenas eget ultrices risus.</P>  
<P>·&#9;Maecenas venenatis tellus id euismod venenatis.</P>  
<P>&#12288;</P></DIR>  
</DIR>  
</DIR>  
</DIR>  
</body>  
</html>
```

# Configure OOBE.xml

10/2/2018 • 4 minutes to read • [Edit Online](#)

To include your registration pages in OOBE, you must configure the appropriate settings of your OOBE.xml file.

A minimum amount of information is required for the registration pages to display. You must provide a page `title`, a page `subtitle`, the `customerinfo` element, at least one additional checkbox **or** one link, and a public key for public/private key encryption.

The following table shows the Oobe.xml elements that correspond to customizable fields on the OEM registration pages:

ELEMENT	SETTING	DESCRIPTION	VALUE
<code>&lt;oem&gt;</code>			
	<code>&lt;registration&gt;</code>	Optional. Additional details are below.	
<code>&lt;registration&gt;</code>			
	<code>&lt;title&gt;</code>	Required if registration element is used. Text to title the registration pages.	String of up to 25 characters.
	<code>&lt;subtitle&gt;</code>	Required if registration element is used. Text to describe the registration pages.	
<code>&lt;customerinfo&gt;</code>			
	<code>&lt;label&gt;</code>	Text to label the top checkbox on registration page two. Required to display the customer information fields on registration page one. Required to display registration pages in OOBE.	String of up to 250 characters. We strongly recommend that you use no more than 100 characters because this length of text will fit on one line.
	<code>&lt;defaultvalue&gt;</code>	Value to set the customerinfo checkbox to selected or not selected.	True or False. True means the check box default condition is selected. False means the check box default condition isn't selected. Default is False.
<code>&lt;checkbox1&gt;</code>			

ELEMENT	SETTING	DESCRIPTION	VALUE
<checkbox2>	<label>	Text to label the second checkbox on registration page two. Required for checkbox1 to appear on registration page two.	String of up to 250 characters. We strongly recommend that you use no more than 100 characters because this length of text will fit on one line.
<checkbox2>	<defaultvalue>	Value to set checkbox1 as selected or not selected.	True or False. True means the check box default condition is selected. False means the check box default condition isn't selected. Default is False.
<checkbox2>			
<checkbox2>	<label>	Text to label the third checkbox on registration page two. Required for checkbox2 to appear on registration page two.	String of up to 250 characters. We strongly recommend that you use no more than 100 characters because this length of text will fit on one line.
<checkbox2>	<defaultvalue>	Value to set checkbox2 as selected or not selected.	True or False. True means the check box default condition is selected. False means the check box default condition isn't selected. Default is False.
<checkbox3>			
<checkbox3>	<label>	Text to label the fourth checkbox on registration page two. Required for checkbox3 to appear on registration page two.	String of up to 250 characters. We strongly recommend that you use no more than 100 characters because this length of text will fit on one line.
<checkbox3>	<defaultvalue>	Value to set checkbox3 as selected or not selected.	True or False. True means the check box default condition is selected. False means the check box default condition isn't selected. Default is False.
<link1>			
<link1>	<label>	Label for the link on registration page one. Required for link1 to appear on registration page one, underneath the four customer information fields.	String of up to 100 characters.

ELEMENT	SETTING	DESCRIPTION	VALUE
	<link>	File must be named linkfile1.html. OOBE searches for these files under the %systemroot%\System32\Oobe\Info folder. OOBE searches for files under the appropriate locale and language specific subfolders of Oobe\Info. Use the <a href="#">HTML sample</a> we provide as a baseline when designing your HTML pages.	linkfile1.html
<link2>			
	<label>	Label for the top link on registration page two. Required for link2 to appear on registration page two.	String of up to 100 characters.
	<link>	File must be named linkfile2.html. OOBE searches for these files under the %systemroot%\System32\Oobe\Info folder. OOBE searches for files under the appropriate locale and language specific subfolders of Oobe\Info. Use the <a href="#">HTML sample</a> we provide as a baseline when designing your HTML pages.	linkfile2.html
<link3>			
	<label>	Label for the second link on registration page two. Required for link3 to appear on registration page two.	String of up to 100 characters.
	<link>	File must be named linkfile3.html. OOBE searches for these files under the %systemroot%\System32\Oobe\Info folder. OOBE searches for files under the appropriate locale and language specific subfolders of Oobe\Info. Use the <a href="#">HTML sample</a> we provide as a baseline when designing your HTML pages.	linkfile3.html

ELEMENT	SETTING	DESCRIPTION	VALUE
<hideSkip>		Optional. Controls whether or not the Skip button is displayed to the user on registration page one.	True or False. True means the skip button is not visible to the user. False means the skip button is displayed as an option to the user. Default is False, resulting in the skip button being visible.

#### NOTE

If you only include one `link` element in your Oobe.xml file, it will appear on registration page one underneath the customer information fields, regardless of which `link` element was used. Similarly, if you only include two `link` elements in your Oobe.xml file, the first will appear on registration page one, and the second will appear as the top link on registration page two.

For example, if you omit `link1` and `link2` from Oobe.xml, and only include `link3`, `link3` will appear underneath the customer information fields on registration page one. If you omit only `link1`, `link2` will appear on registration page one, and `link3` will appear as the top link on registration page two.

For more information on these settings, and the others you can configure, please see [Oobe.xml Settings](#).

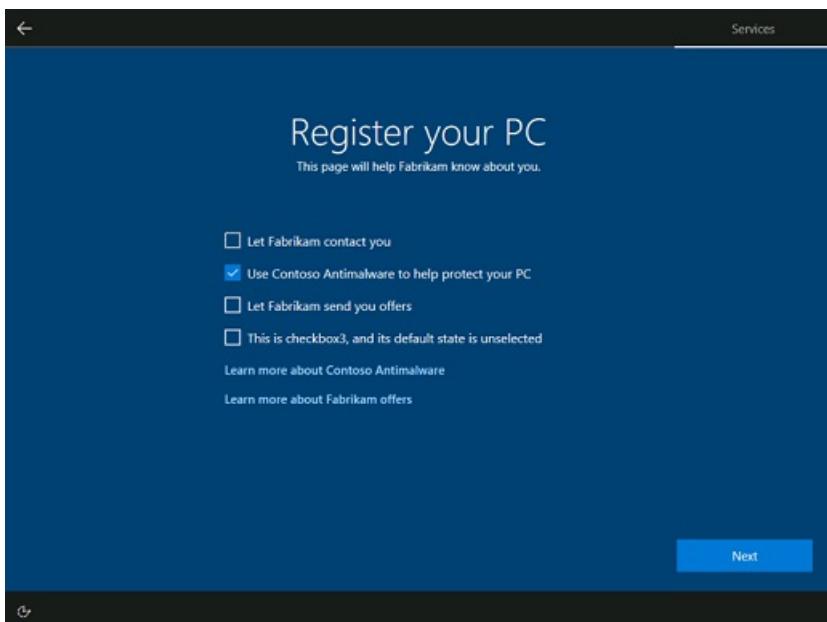
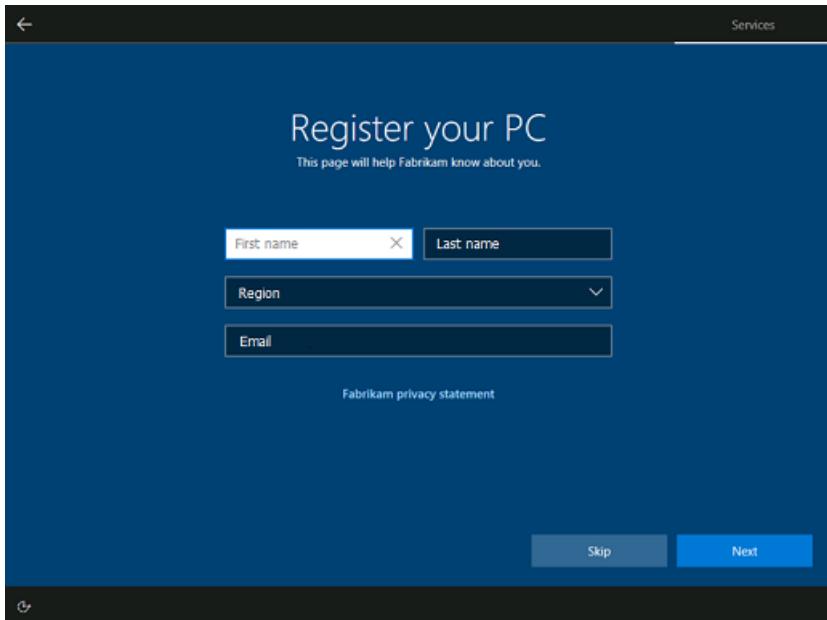
## XML example

```

<oobe>
  <oem>
    <registration>
      <title>Register your PC</title>
      <subtitle>This page will help Fabrikam know about you.</subtitle>
      <customerinfo>
        <label>Let Fabrikam contact you</label>
      </customerinfo>
      <checkbox1>
        <label>Use Contoso Antimalware to help protect your PC</label>
        <defaultValue>true</defaultValue>
      </checkbox1>
      <checkbox2>
        <label>Let Fabrikam send you offers</label>
      </checkbox2>
      <checkbox3>
        <label>This is checkbox3, and its default state is unselected</label>
      </checkbox3>
      <link1>
        <label>Fabrikam privacy statement</label>
      </link1>
      <link2>
        <label>Learn more about Contoso Antimalware</label>
      </link2>
      <link3>
        <label>Learn more about Fabrikam offers</label>
      </link3>
      <hideSkip>false</hideSkip>
    </registration>
  </oem>
</oobe>

```

Here are the OEM registration pages that will appear as a result of the XML example above:



# Protect and collect user data

10/2/2018 • 9 minutes to read • [Edit Online](#)

If a customer enters information into the OEM registration pages, the following files are created when they complete OOBE:

- **Userdata.blob**. An encrypted XML file that contains all the values in all user-configurable elements on the registration pages, including customer information fields and checkbox states.
- **SessionKey.blob**. Generated during encryption of Userdata.blob. Contains a session key needed for the decryption process.
- **Userchoices.xml**. An un-encrypted XML file that contains the checkbox labels and values for all checkboxes included on the registration pages.

## NOTE

If a customer clicks `skip` on the first registration page, no data is written or stored to these files, not even the checkbox default states.

The timestamp of the user's out of box experience is also added to the Windows registry under this key:

```
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\OOBE\Stats [EndTimeStamp]
```

This registry value is created regardless of whether the registration pages are included in OOBE. The timestamp is written in UTC (Coordinated Universal Time) format; specifically, it is a `SYSTEMTIME` value written as a serialized blob of data to the registry.

In order for you to access and use the customer information, take the following steps:

1. [Generate a public/private key pair](#), and place the public key in the `%systemroot%\system32\Oobe\Info` folder of the image.
2. [Collect the encrypted customer data](#) using an app or a service that runs roughly 30 minutes after the first logon completes.
3. [Send the data to your server for decryption](#) using SSL. You can then decrypt the session key to decrypt the customer data.

## Generate a public/private key pair

To protect customer data, you must generate a public/private key pair, and the public key must be placed in the `%systemroot%\system32\Oobe\Info` folder. If you're deploying images to multiple regions or in multiple languages, you should put the public key directly under region and language-specific subdirectories, following the same rules as you would for region or language-specific Oobe.xml files as described in [How Oobe.xml works](#).

## IMPORTANT

You must never place the private key on the customer's PC. Instead, it should be stored securely on your servers so the data can be decrypted after it's uploaded. If a customer clicks Next on the Registration pages, Windows uses the public key to create Sessionkey.blob in the `%systemroot%\system32\Oobe\Info` folder. Your service or Microsoft Store app should upload the data to your server by using SSL. You then need to decrypt the session key to decrypt the customer data.

If there's no public key in the `%systemroot%\system32\Oobe\Info` folder, the registration pages aren't shown.

## Generate public and private keys

Make this sequence of calls to generate the public and private keys.

1. Acquire crypt context using the [CryptAcquireContext API](#). Provide these values:

- `pszProvider` is `MS_ENH_RSA_AES_PROV`
- `dwProvType` is `PROV_RSA_AES`

2. Generate RSA crypt key using the [CryptGenKey API](#). Provide these values:

- `AlgId` is `CALG_RSA_KEYX`
- `dwFlags` is `CRYPT_EXPORTABLE`

3. Serialize the public key portion of the crypt key from Step 2 using the [CryptExportKey API](#). Provide this value:

- `dwBlobType` is `PUBLICKEYBLOB`

4. Write the serialized public key bytes from Step 3 to file Pubkey.blob using the standard [Windows File Management functions](#).

5. Serialize the private key portion of the crypt key from Step 2 using the [CryptExportKey API](#). Provide this value

- `dwBlobType` is `PRIVATEKEYBLOB`

6. Write the serialized private key bytes from step 5 to file Prvkey.blob using the standard Windows File API.

## Code snippet

This code snippet shows how to generate the keys:

```
HRESULT CryptExportKeyHelper(_In_ HCRYPTKEY hKey, _In_opt_ HCRYPTKEY hExpKey, DWORD dwBlobType,
                            _Outptr_result_bytebuffer_(*pcbBlob) BYTE **ppbBlob, _Out_ DWORD *pcbBlob);

HRESULT WriteByteArrayToFile(_In_ PCWSTR pszPath, _In_reads_bytes_(cbData) BYTE const *pbData, DWORD cbData);

// This method generates an OEM public and private key pair and writes it to Pubkey.blob and Prvkey.blob
HRESULT GenerateKeysToFiles()
{
    // Acquire crypt provider. Use provider MS_ENH_RSA_AES_PROV and provider type PROV_RSA_AES to decrypt the
    // blob from OOBE.
    HCRYPTPROV hProv;
    HRESULT hr = CryptAcquireContext(&hProv, L"OEMDecryptContainer", MS_ENH_RSA_AES_PROV,
                                    PROV_RSA_AES, CRYPT_NEWKEYSET) ? S_OK : HRESULT_FROM_WIN32(GetLastError());
    if (hr == NTE_EXISTS)
    {
        hr = CryptAcquireContext(&hProv, L"OEMDecryptContainer", MS_ENH_RSA_AES_PROV,
                                PROV_RSA_AES, 0) ? S_OK : HRESULT_FROM_WIN32(GetLastError());
    }

    if (SUCCEEDED(hr))
    {
        // Call CryptGenKey to generate the OEM public and private key pair. OOBE expects the algorithm to be
        // CALG_RSA_KEYX.
        HCRYPTKEY hKey;
        hr = CryptGenKey(hProv, CALG_RSA_KEYX, CRYPT_EXPORTABLE, &hKey) ? S_OK :
        HRESULT_FROM_WIN32(GetLastError());
        if (SUCCEEDED(hr))
        {
            // Call CryptExportKeyHelper to serialize the public key into bytes.
            BYTE *pbPubBlob;
            DWORD cbPubBlob;
            hr = CryptExportKeyHelper(hKey, NULL, PUBLICKEYBLOB, &pbPubBlob, &cbPubBlob);
            if (SUCCEEDED(hr))
            {
                // Call CryptExportKey again to serialize the private key into bytes.
            }
        }
    }
}
```

```

        BYTE *pbPrvBlob;
        DWORD cbPrvBlob;
        hr = CryptExportKeyHelper(hKey, NULL, PRIVATEKEYBLOB, &pbPrvBlob, &cbPrvBlob);
        if (SUCCEEDED(hr))
        {
            // Now write the public key bytes into the file pubkey.blob
            hr = WriteByteArrayToFile(L"pubkey.blob", pbPubBlob, cbPubBlob);
            if (SUCCEEDED(hr))
            {
                // And write the private key bytes into the file Prvkey.blob
                hr = WriteByteArrayToFile(L"prvkey.blob", pbPrvBlob, cbPrvBlob);
            }
            HeapFree(GetProcessHeap(), 0, pbPrvBlob);
        }
        HeapFree(GetProcessHeap(), 0, pbPubBlob);
    }
    CryptDestroyKey(hKey);
}
CryptReleaseContext(hProv, 0);
}
return hr;
}

HRESULT CryptExportKeyHelper(_In_ HCRYPTKEY hKey, _In_opt_ HCRYPTKEY hExpKey, DWORD dwBlobType,
_Inptr_result_bytebuffer_(*pcbBlob) BYTE **ppbBlob, _Out_ DWORD *pcbBlob)
{
    *ppbBlob = nullptr;
    *pcbBlob = 0;

    // Call CryptExportKey the first time to determine the size of the serialized key.
    DWORD cbBlob = 0;
    HRESULT hr = CryptExportKey(hKey, hExpKey, dwBlobType, 0, nullptr, &cbBlob) ? S_OK :
HRESULT_FROM_WIN32(GetLastError());
    if (SUCCEEDED(hr))
    {
        // Allocate a buffer to hold the serialized key.
        BYTE *pbBlob = reinterpret_cast<BYTE *>(CoTaskMemAlloc(cbBlob));
        hr = (pbBlob != nullptr) ? S_OK : E_OUTOFMEMORY;
        if (SUCCEEDED(hr))
        {
            // Now export the key to the buffer.
            hr = CryptExportKey(hKey, hExpKey, dwBlobType, 0, pbBlob, &cbBlob) ? S_OK :
HRESULT_FROM_WIN32(GetLastError());
            if (SUCCEEDED(hr))
            {
                *ppbBlob = pbBlob;
                *pcbBlob = cbBlob;
                pbBlob = nullptr;
            }
            CoTaskMemFree(pbBlob);
        }
    }
    return hr;
}

HRESULT WriteByteArrayToFile(_In_ PCWSTR pszPath, _In_reads_bytes_(cbData) BYTE const *pbData, DWORD cbData)
{
    bool fDeleteFile = false;
    HANDLE hFile = CreateFile(pszPath, GENERIC_WRITE, 0, nullptr, CREATE_ALWAYS, FILE_ATTRIBUTE_NORMAL,
nullptr);
    HRESULT hr = (hFile == INVALID_HANDLE_VALUE) ? HRESULT_FROM_WIN32(GetLastError()) : S_OK;
    if (SUCCEEDED(hr))
    {
        DWORD cbWritten;
        hr = WriteFile(hFile, pbData, cbData, &cbWritten, nullptr) ? S_OK :
HRESULT_FROM_WIN32(GetLastError());
        fDeleteFile = FAILED(hr);
        CloseHandle(hFile);
    }
}

```

```
    if (fDeleteFile)
    {
        DeleteFile(pszPath);
    }
    return hr;
}
```

## Collect encrypted customer data

Create and preinstall a Microsoft Store app, or write a service to run after first sign-in, to:

1. Collect the encrypted customer data, including the user name from the [Windows.System.User namespace](#), as well as the local time stamp of first sign-in.
2. Upload that data set to your server for decryption and use.

To use a Microsoft Store app to collect the data, assign its Application User Model ID (AUMID) to the [Microsoft-Windows-Shell-Setup | OOBE | OEMAppId](#) Unattend setting. Windows will pass the timestamp, user data, session key, and checkbox state data to the application data folder for the OEM app, that is associated with the first user to logon to the device. For example, `%localappdata%\packages\[OEM app package family name]\LocalState` for that user.

If you create and run a service to upload the data, you should set the service to run at least 30 minutes after the user gets to the Start screen, and only run the service once. Setting your service to run at this time ensures that your service won't consume system resources in the background while users are getting their first chance to explore the Start screen and their apps. The service must gather the data from within the OOBE directory, as well as the time stamp and user name, as applicable. The service should also determine what actions to take in response to the user's choices. For example, if the user opted in to an anti-malware app trial, your service should start the trial rather than rely on the anti-malware app to decide if it should run. Or, as another example, if your user opted in to emails from your company or partner companies, your service should communicate that info to whomever handles your marketing emails.

For more info about how to write a service, see [Developing Windows Service Applications](#).

## Send data to your server for decryption

Your service or Microsoft Store app should upload the data to your server using SSL. You then need to decrypt the session key to decrypt the customer data.

### Decrypt the data

Make this sequence of calls to decrypt the data:

1. Acquire crypt context by using the [CryptAcquireContext API](#). Provide these values:
  - `pszProvider` is `MS_ENH_RSA_AES_PROV`
  - `dwProvType` is `PROV_RSA_AES`
2. Read the OEM private key file (`Prvkey.blob`) from disk using the standard Windows File API.
3. Convert the private key bytes into a crypt key using the [CryptImportKey API](#).
4. Read the OOBE-generated session key file (**Sessionkey.blob**) from disk using the standard Windows File API.
5. Use the private key from Step 3 to convert the session key bytes into a crypt key, using the [CryptImportKey API](#).
6. Export key (`hPubKey`) is the private key imported in Step 3.
7. Read OOBE-written encrypted user data (**Userdata.blob**) from disk using the standard Windows File API.
8. Use session key (from Step 5) to decrypt the user data, using [CryptDecrypt](#).

### Code snippet

This code snippet shows how to decrypt the data:

```
HRESULT DecryptHelper(_In_reads_bytes_(cbData) BYTE *pbData, DWORD cbData, _In_ HCRYPTKEY hPrvKey,
_Outptr_result_bytebuffer_(*pcbPlain) BYTE **ppbPlain, _Out_ DWORD *pcbPlain);
HRESULT ReadFileToByteArray(_In_ PCWSTR pszPath, _Outptr_result_bytebuffer_(*pcbData) BYTE **ppbData, _Out_
DWORD *pcbData);

// This method uses the specified Userdata.blob (pszDataFilePath), Sessionkey.blob (pszSessionKeyPath), and
Prvkey.blob (pszPrivateKeyPath)
// and writes the plaintext XML user data to Plaindata.xml
HRESULT UseSymmetricKeyFromFileToDecrypt(_In_ PCWSTR pszDataFilePath, _In_ PCWSTR pszSessionKeyPath, _In_
PCWSTR pszPrivateKeyPath)
{
    // Acquire crypt provider. Use provider MS_ENH_RSA_AES_PROV and provider type PROV_RSA_AES to decrypt the
blob from OOBE.
    HCRYPTPROV hProv;
    HRESULT hr = CryptAcquireContext(&hProv, L"OEMDecryptContainer", MS_ENH_RSA_AES_PROV, PROV_RSA_AES,
CRYPT_NEWSKEYSET) ? S_OK : HRESULT_FROM_WIN32(GetLastError());
    if (hr == NTE_EXISTS)
    {
        hr = CryptAcquireContext (&hProv, L"OEMDecryptContainer", MS_ENH_RSA_AES_PROV, PROV_RSA_AES, 0) ? S_OK
: HRESULT_FROM_WIN32(GetLastError());
    }

    if (SUCCEEDED(hr))
    {
        // Read in the OEM private key file.
        BYTE *pbPrvBlob;
        DWORD cbPrvBlob;
        hr = ReadFileToByteArray(pszPrivateKeyPath, &pbPrvBlob, &cbPrvBlob);
        if (SUCCEEDED(hr))
        {
            // Convert the private key file bytes into an HCRYPTKEY.
            HCRYPTKEY hKey;
            hr = CryptImportKey(hProv, pbPrvBlob, cbPrvBlob, 0, 0, &hKey) ? S_OK :
HRESULT_FROM_WIN32(GetLastError());
            if (SUCCEEDED(hr))
            {
                // Read in the encrypted session key generated by OOBE.
                BYTE *pbSymBlob;
                DWORD cbSymBlob;
                hr = ReadFileToByteArray(pszSessionKeyPath, &pbSymBlob, &cbSymBlob);
                if (SUCCEEDED(hr))
                {
                    // Convert the encrypted session key file bytes into an HCRYPTKEY.
                    // This uses the OEM private key to decrypt the session key file bytes.
                    HCRYPTKEY hSymKey;
                    hr = CryptImportKey(hProv, pbSymBlob, cbSymBlob, hKey, 0, &hSymKey) ? S_OK :
HRESULT_FROM_WIN32(GetLastError());
                    if (SUCCEEDED(hr))
                    {
                        // Read in the encrypted user data written by OOBE.
                        BYTE *pbCipher;
                        DWORD dwCipher;
                        hr = ReadFileToByteArray(pszDataFilePath, &pbCipher, &dwCipher);
                        if (SUCCEEDED(hr))
                        {
                            // Use the session key to decrypt the encrypted user data.
                            BYTE *pbPlain;
                            DWORD dwPlain;
                            hr = DecryptHelper(pbCipher, dwCipher, hSymKey, &pbPlain, &dwPlain);
                            if (SUCCEEDED(hr))
                            {
                                hr = WriteByteArrayToFile(L"plaindata.xml", pbPlain, dwPlain);
                                HeapFree(GetProcessHeap(), 0, pbPlain);
                            }
                            HeapFree(GetProcessHeap(), 0, pbCipher);
                        }
                    }
                }
            }
        }
    }
}
```

```

        CryptDestroyKey(hSymKey);
    }
    HeapFree(GetProcessHeap(), 0, pbSymBlob);
}
else if (hr == HRESULT_FROM_WIN32(ERROR_FILE_NOT_FOUND))
{
    wcout << L"Couldn't find session key file [" << pszSessionKeyPath << L"]" << endl;
}
CryptDestroyKey(hKey);
}
HeapFree(GetProcessHeap(), 0, pbPrvBlob);
}
else if (hr == HRESULT_FROM_WIN32(ERROR_FILE_NOT_FOUND))
{
    wcout << L"Couldn't find private key file [" << pszPrivateKeyPath << L"]" << endl;
}
CryptReleaseContext(hProv, 0);
}
return hr;
}

HRESULT DecryptHelper(_In_reads_bytes_(cbData) BYTE *pbData, DWORD cbData, _In_ HCRYPTKEY hPrvKey,
_Outptr_result_bytebuffer_(*pcbPlain) BYTE **ppbPlain, _Out_ DWORD *pcbPlain)
{
    BYTE *pbCipher = reinterpret_cast<BYTE *>(HeapAlloc(GetProcessHeap(), 0, cbData));
    HRESULT hr = (pbCipher != nullptr) ? S_OK : E_OUTOFMEMORY;
    if (SUCCEEDED(hr))
    {
        // CryptDecrypt will write the actual length of the plaintext to cbPlain.
        // Any block padding that was added during CryptEncrypt won't be counted in cbPlain.
        DWORD cbPlain = cbData;
        memcpy(pbCipher, pbData, cbData);
        hr = ResultFromWin32Bool(CryptDecrypt(hPrvKey,
                                               0,
                                               TRUE,
                                               0,
                                               pbCipher,
                                               &cbPlain));
    }
    if (SUCCEEDED(hr))
    {
        *ppbPlain = pbCipher;
        *pcbPlain = cbPlain;
        pbCipher = nullptr;
    }
    HeapFree(GetProcessHeap(), 0, pbCipher);
}
return hr;
}

HRESULT ReadFileToByteArray(_In_ PCWSTR pszPath, _Outptr_result_bytebuffer_(*pcbData) BYTE **ppbData, _Out_
DWORD *pcbData)
{
    *ppbData = nullptr;
    *pcbData = 0;
    HANDLE hFile = CreateFile(pszPath, GENERIC_READ, 0, nullptr, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL,
    nullptr);
    HRESULT hr = (hFile == INVALID_HANDLE_VALUE) ? HRESULT_FROM_WIN32(GetLastError()) : S_OK;
    if (SUCCEEDED(hr))
    {
        DWORD cbSize = GetFileSize(hFile, nullptr);
        hr = (cbSize != INVALID_FILE_SIZE) ? S_OK : ResultFromKnownLastError();
        if (SUCCEEDED(hr))
        {
            BYTE *pbData = reinterpret_cast<BYTE *>(CoTaskMemAlloc(cbSize));
            hr = (pbData != nullptr) ? S_OK : E_OUTOFMEMORY;
            if (SUCCEEDED(hr))
            {
                DWORD cbRead;
                hr = ReadFile(hFile, pbData, cbSize, &cbRead, nullptr) ? S_OK :
HRESUIT FROM WIN32(GetLastError());

```

```
    if (SUCCEEDED(hr))
    {
        *ppbData = pbData;
        *pcbData = cbSize;
        pbData = nullptr;
    }
    CoTaskMemFree(pbData);
}
CloseHandle(hFile);
}
return hr;
}
```

# Automate OOB

10/2/2018 • 2 minutes to read • [Edit Online](#)

You can use Unattend settings to prevent some or all of the user interface (UI) pages from appearing in Windows OOB.

To learn more about creating an answer file using Unattend, as well as a full list of Unattend settings available to you, see the [Unattended Windows Setup Reference](#).

SETTING	CONFIGURATION PASS	DESCRIPTION	APPLIES TO
Microsoft-Windows-International-Core settings: <a href="#">InputLocale</a> , <a href="#">SystemLocale</a> , <a href="#">UILanguage</a> , and <a href="#">UserLocale</a> .	oobeSystem	Specifies the region-specific defaults of the Windows installation.	Windows 10 for desktop editions (Home, Pro, Enterprise, and Education) and Windows Server 2016
Microsoft-Windows-Shell-Setup/UserAccounts	oobeSystem	Specifies the user accounts, and passwords, to create on the Windows installation. The account can be a user account, a domain account, or the default administrator account.	Windows 10 for desktop editions and Windows Server 2016
Microsoft-Windows-Shell-Setup/OOBE settings: <a href="#">HideEULAPage</a> , <a href="#">HideOEMRegistrationScreen</a> , <a href="#">HideOnlineAccountScreens</a> , <a href="#">HideWirelessSetupInOOBE</a> , and <a href="#">HideLocalAccountScreen</a> .	oobeSystem	Specifies whether certain OOBE screens will be hidden.	Windows 10 for desktop editions and Windows Server 2016
Microsoft-Windows-Shell-Setup/OOBE/ProtectYourPC	oobeSystem	Specifies whether your device is configured with Express settings, such as sending data to Microsoft, letting Windows and apps request the user's localization, and turning on protection against malicious web content.	Windows 10 for desktop editions and Windows Server 2016

## Related topics

[Automate Windows Setup](#)

[Windows Auto Pilot](#)

# Customize the Retail Demo Experience (RDX)

10/11/2018 • 20 minutes to read • [Edit Online](#)

Showcase your new devices on the retail sales floor with a rich, engaging experience with the Windows Retail Demo Experience (RDX).

- **Attract shoppers with demo videos.** We've included videos that show off the latest Windows 10 features. Add your own videos to show off your own unique hardware, apps, and services.
- **Let shoppers try it out.** Shoppers can experience the device first-hand, working with sample data in contacts, photos, email and messaging apps.

Retail mode works best when demo devices have high-speed Internet access.

Customizable components of RDX:

- **Attract loop app:** a perpetually looping video or images intended to attract customers to the device. The content is intended to draw the customer in to interact with the device.
- **Retail Demo app:** an app that launches automatically when a customer ends the attract loop by tapping a keyboard key, clicking the mouse, or touching the screen (if touchscreen) while the Attract loop app plays. The Retail Demo app educates the customer about the device, Windows, and associated services available with the purchase of the device. After a period of inactivity, the attract loop begins playing again.
- **Demo mode content:** content the customer can interact with during the demo. This includes pre-loaded (image) or downloaded app content, documents, music, photos, videos, and Store apps.
- **Setup and operation of retail demo mode:** determine RDX enablement on the device, automatic device clean-up between customers, and automated removal of RDX content after purchase.
- **Digital Fact Tag app:** an app that launches automatically at the same time as the Retail Demo app. This app sits on the right side of the screen and displays key information in a perpetual way for the shopper. The app cannot be closed by the shopper, nor do apps display above or behind the app.

## RDX 3.0

RDX 3.0 is included in Windows 10, version 1809. For Windows 10, version 1803, you can preload RDX 2.0, and once the device is connected, it will upgrade to RDX 3.0 automatically.

Key updates include:

- **The Retail Demo app has a new webpage-style layout.** New home page, navigation style, and content.
- **New: RD Provisioning extension API** allows you to manage online assets yourself. In RDX 2.0, online assets are managed through the Retail Demo Asset Manager (RDAM), and the time from start to finish (submission > review > approval > sent to devices) is 2-3 weeks. If you manage your own online assets using our API, you may be able to complete these tasks faster.
- **New: On-device admin (ODA) app** (part of the provisioning API) allows retailers to update specs, price locally on non-networked devices.
- **Coming soon: Digital fact tag (DFT)** shows customers device specs and price. This feature will be available as part of an online update. After receiving the online update, retailers can manually update the DFT through the Retail Demo Mode Advanced Configuration menus. To learn more, see the [RDX Windows Experience Guide \(WEG\)](#).

## Attract loop

The retail demo experience begins with a video, which plays repeatedly while the device is idle. When the video attract loop plays, the Start screen is restored back to a pre-set state. Photos and videos taken by previous customers are deleted and the demo photos are also restocked.

#### **IMPORTANT**

The device must be plugged into AC power for the video attract loop to start.

### **Design recommendations**

Create your own custom attraction video that highlights key features of your device.

Use full-screen imagery to focus on key selling points (KSPs) of the device. Our research shows that shoppers are attracted to loops that show off hardware features with fast moving graphics and colorful imagery, but loops that function as advertisements don't resonate with shoppers.

Limit the video message to 1 or 2 KSPs. The loop is designed to get shoppers to interact with device. If the attract loop looks like an advertisement, shoppers are less likely to pay attention and interact with the system.

Avoid text in your video. Videos without text are easier to scale across multiple markets and locales since there are no localization costs. In addition, shoppers typically view only part of the video, so your text might not be viewed in its entirety.

We strongly recommend that you use the attract loop to show how your brand, apps, services, and the devices themselves are differentiated from your competitors.

### **Requirements**

The following are the specifications for the attract loop video.

- Videos must be less than 60 seconds.
- Must not include an audio track.
- Acceptable compression format: H.264/MPEG4
- Videos must be designed so they don't cause screen burn even when played for hours at a time for the entire shelf life of the device.
- The source video should be appropriate quality to ensure the best possible playback on the device based on its graphics rendering capabilities (resolution, color capabilities, and graphics processing power).
- We recommend matching the video resolution to the optimal resolution on each device when possible.  
Otherwise, resolution should be 1920 x 1080.

### **Add the video to the image**

You can replace the default attract loop video with your own customized video, which you can save to your device images. Doing so makes this video available to the shopper even if the retailer never connects the demo device to the internet. This content should not be time-sensitive or seasonal, and it should be appropriate for all regions and languages the device will ship to.

Create the default set of content first. This content is stored in the \Neutral file path, and it must be named:  
**attract.wmv:**

- %programdata%\Microsoft\Windows\RetailDemo\OfflineContent\OEM\Content\Neutral\AttractContent\attract.wmv

**For devices sold in multiple regions or with multiple languages:** You can add region and/or language-specific versions for attract loops. When there is no region-specific or language-specific content, the default (\Neutral) content is displayed.

For a complete list of supported languages and locales, see [Language Identifier Constants and Strings](#).

- %programdata%\Microsoft\Windows\RetailDemo\OfflineContent\OEM\Content\[locale]\AttractContent\attract.wmv

- %programdata%\Microsoft\Windows\RetailDemo\OfflineContent\OEM\[region]\Content\Neutral\AttractContent\attract.wmv
- %programdata%\Microsoft\Windows\RetailDemo\OfflineContent\OEM\[region]\Content\[locale]\AttractContent\attract.wmv

Example: Canada-specific content in French:

- %programdata%\Microsoft\Windows\RetailDemo\OfflineContent\OEM\CA\Content\fr-ca\AttractContent\attract.wmv

### Add the video using the Microsoft RDX Submission Tool

Add your default attract loop video, as well as any updated videos, to the [RDX Submission Tool](#). If you don't currently have an account for the RDX Submission Tool, please reach out to your Account Manager, and let them know which Microsoft Account (MSA) you'd like to use to access the tool.

In the tool, you can target your videos by language, region, and model, so that when a targeted device is connected to the internet, it automatically replaces the video and plays it.

## Retail Demo app

To get started, you can include the Windows inbox Retail Demo app.

In RDX 2.0, shoppers select content through navigation tabs:



*Example of current RDX 2.0 Navigation Tabs experience*

In RDX 3.0, shoppers select content through tiles, and can see more info on the Digital Fact Tag (right):



Each content page contains one or more sections that are comprised of media (images and video), text copy, and Call-to-Action (CTA) buttons or links to encourage the shopper to explore the featured content. If a content page contains multiple sections, a feature bar displays at the bottom. The customer can move between content sections of the page by selecting features in the bar or by scrolling up and down the page.

The app highlights key features as determined by the Microsoft marketing team. Some content in the app only appears when Office is installed or certain hardware is detected. For example, when Office 365 is installed, the Office section may show a "Try it now" button to open Word or PowerPoint. When a digitizer is detected, the Office section may display a collection of videos showing how Office works with a pen.

The content shown changes based on the device. For example, if you have preinstalled Office 365, the demos show Office's pen and inking functionality.

When a shopper closes the Retail Demo app, they see the desktop of the device.

In RDX 3.0, they'll also see a Digital Fact Tag to the right (unless the language reads right-to-left, in which case the Digital Fact Tag is on the left).

### Create custom content

OEMs and Retailers can create new custom content for the Retail Demo app using the [Microsoft RDX Submission Tool](#). If you don't currently have an account for the RDX Submission Tool, please reach out to your Account Manager, and let them know which Microsoft Account (MSA) you'd like to use to access the tool.

In the tool, you can create a base set of content which you can save to your device images. Doing so makes this content available to the shopper even if the retailer never connects the demo device to the internet. This content should not be time-sensitive or seasonal, and it should be appropriate for all regions and languages the device will ship to.

In the tool you can also create new or updated content which can be delivered online to your devices. This content can be targeted by language, region, and model, so that when a targeted device is connected to the internet, it automatically receives the updates and shows the new content.

OEMs can specify a theme color, navigation selected-button color, and logos for the Retail Demo app, in addition to adding unique page content. Colors and logos are specified at the app level, and content is specified at the page level. OEMs can also choose between one of three templates: Hero, Immersive Hero, and Mosaic (described below).

### Template options, examples, and requirements

There are four templates available for the Retail Demo app content sections: Hero, Feature, Immersive Hero, and Mosaic.

#### Hero template

Use the Hero template to showcase key content. A full-bleed image must be used – this can be combined with a title, copy, and/or call-to-action link. This template can also support full-bleed videos. All text in a video must be embedded into the media. Below are the media and copy requirements for the Hero template:

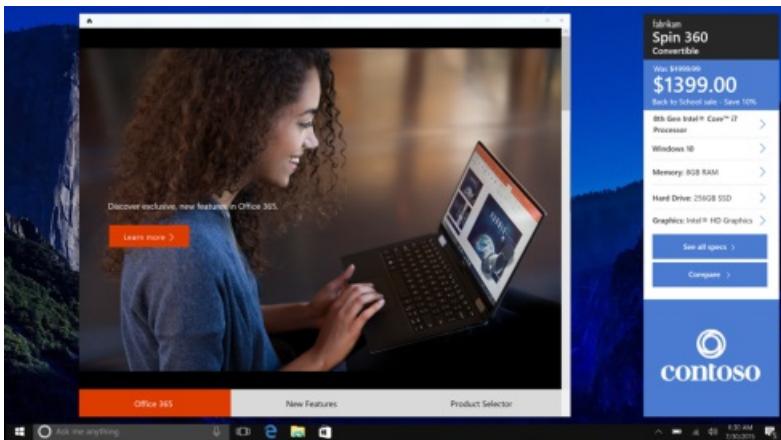
ELEMENT	REQUIREMENTS
Image or video	Images must be PNG, have a transparent background and no padding; videos must be .mp4 files. Resolution recommendations are provided by the RDX Submission Tool when you build your content and vary by template.
Logo	946 x 220 pixels; images must be PNG, have a transparent background and no padding.
Header	We recommend a 55-character limit, not counting spaces in between characters. This allows room for localization.
Sub header	We recommend a 55-character limit, not counting spaces in between characters. This allows room for localization.

ELEMENT	REQUIREMENTS
Paragraph	140 characters, not counting spaces in between characters.
Call-to-action (CTA) button text	We recommend an 11-character limit, not counting spaces in between characters.
Legal text	150 characters, not counting spaces in between characters.

There are 4 actions that can be set for a CTA button:

1. Jump to another page within group
2. Launch an app
3. Launch the default browser and go to a URL (online devices only)
4. Open media (image, video, or document)

Here is an example of a Hero template:



#### Immersive Hero template

Use the Immersive Hero template when there is a specific part of the device, or a specific feature, you want to call out. You can switch the position of the copy and the image for an alternative placement. A combination of title, copy, and/or a call-to-action link can be used here.

Below are the media and copy requirements for the Immersive Hero template:

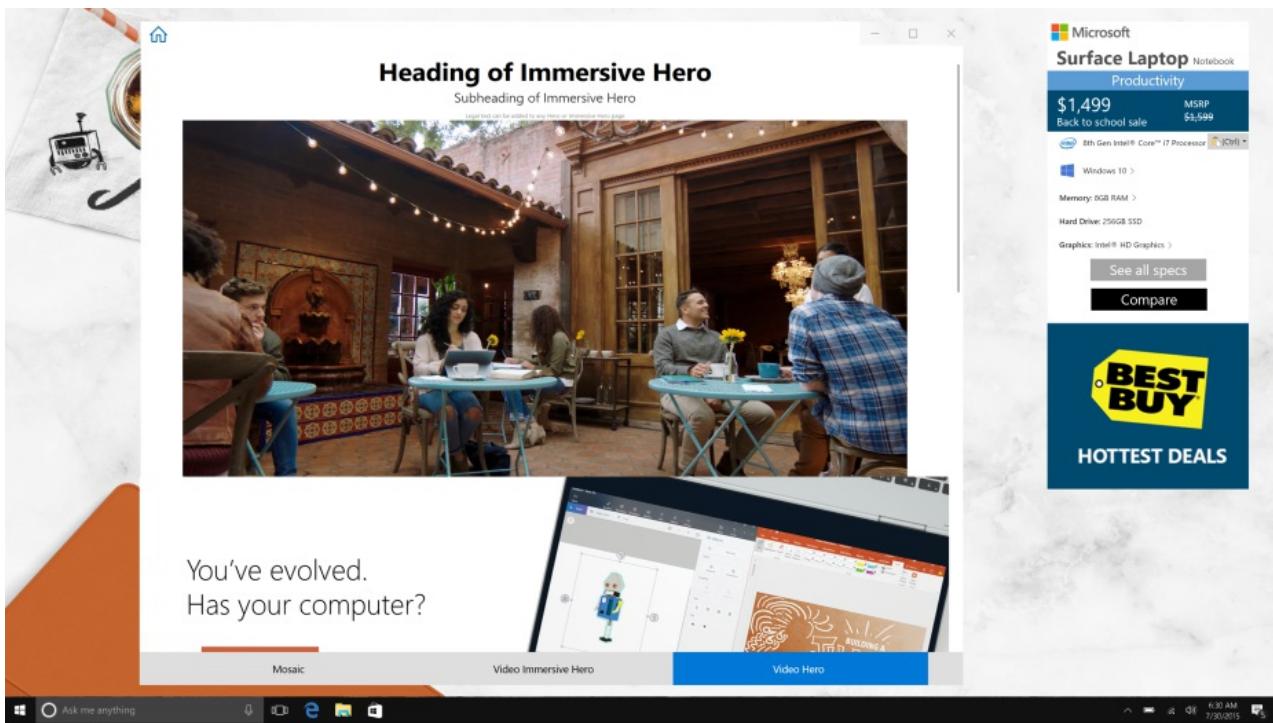
ELEMENT	REQUIREMENTS
Image or video	Images must be PNG, have a transparent background and no padding; videos must be .mp4 files. Resolution recommendations are provided by the RDX Submission Tool when you build your content and vary by template.
Logo	946 x 220 pixels; images must be PNG, have a transparent background and no padding.
Header	We recommend a 55-character limit, not counting spaces in between characters. This allows room for localization.
Sub header	We recommend a 55-character limit, not counting spaces in between characters. This allows room for localization.
Paragraph	140 characters, not counting spaces in between characters.

ELEMENT	REQUIREMENTS
Call-to-action (CTA) button text	We recommend an 11-character limit, not counting spaces in between characters.
Legal text	150 characters, not counting spaces in between characters.

There are 4 actions that can be set for a CTA button:

1. Jump to another page within group
2. Launch an app
3. Launch the default browser and go to a URL (online devices only)
4. Open media (image, video, or document)

Below is an example of the Immersive Hero template.



## Mosaic template

Use this template to show components as a graphic montage. This template is very versatile because you can use it with or without a text file. It always extends the full content area. Use the mosaic to support a theme, communicate an idea, or present top picks around particular topics. Add call-to-action links if you need to direct users to another page. CTAs are typically centrally aligned and appear at the bottom of the tile.

The mosaic layout follows several predefined patterns, depending on the number of tiles you wish to include. The layout will appear as follows:

### 2 Tiles | RDX 2.0



### 3 Tiles | RDX 2.0



### 4 Tiles | RDX 2.0



### 6 Tiles | Coming



### 7 Tiles | Coming



### 8 Tiles | Coming



Below are the media and copy requirements for the Mosaic template:

ELEMENT	REQUIREMENTS
Mosaic layout	Mosaic template allows for 2 – 4 tiles. The layout of the tiles is auto-generated based on number of tiles input into the RDX Submission Tool. The layout is not adjustable, so you will need to place tiles in the correct order for the layout.
Tile background image	Resolution varies based on number of tiles used, but generally images should be square or 2:1 resolution appropriate to the screen; images must be PNG, have a transparent background and no padding
Tile paragraph image	Recommend 220 x 220 pixels; images must be PNG, have a transparent background and no padding
Logo	946 x 220 pixels; images must be PNG, have a transparent background and no padding.
Header	We recommend a 55-character limit, not counting spaces in between characters. This allows room for localization.
Sub header	We recommend a 55-character limit, not counting spaces in between characters. This allows room for localization.
Paragraph	140 characters, not counting spaces in between characters.

ELEMENT	REQUIREMENTS
Call-to-action (CTA) button text	We recommend an 11-character limit, not counting spaces in between characters.
Legal Text	Mosaic does not support legal text on tiles due to tile size.

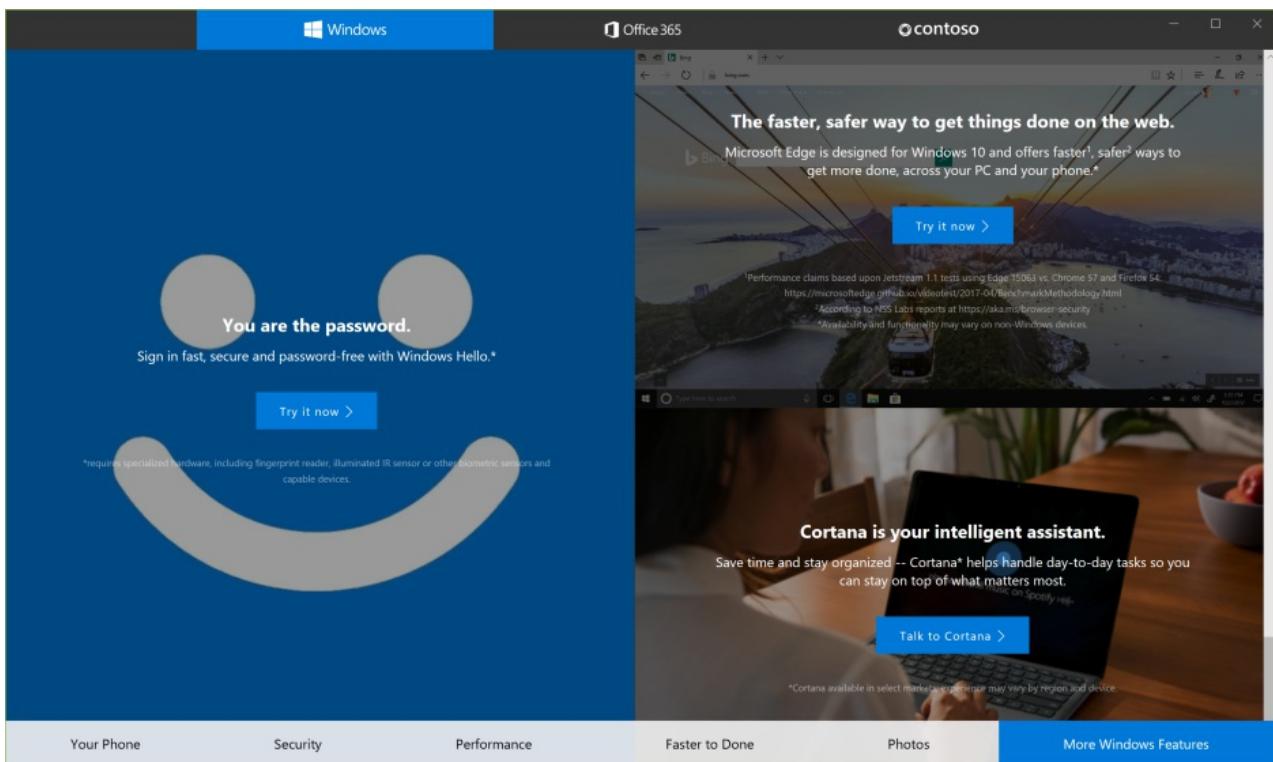
There are 4 actions that can be set for a CTA button:

1. Jump to another page within group
2. Launch an app
3. Launch the default browser and go to a URL
4. Open media (image, video, or document)

#### NOTE

Keep in mind that if you are building content for offline devices, CTA buttons should not open URLs as this will create a poor user experience.

Here is an example of a 3-tile Mosaic layout:



#### Add Retail Demo app content packages to OEM image

You will need to go through the process outlined below for each language you intend to provide Retail Demo app content for. The process will take you through building content for specific regions or locales, and storing it in the appropriate file path.

Create the default set of content first. This content should be appropriate for all regions and languages the device will ship to. This content is stored in the \Neutral file path:

#### For devices sold in multiple regions:

You can add region and/or language-specific versions for attract loops. When there is no region-specific or language-specific content, the default (\Neutral) content is displayed.

For a complete list of supported languages and locales, see [Language Identifier Constants and Strings](#).

**NOTE**

To support languages, your images will need Retail Demo Mode language packs.

**Add the content to the image:**

1. After building your content on the [RDX Submission Tool](#), download and save locally.
2. Save the file as **`oem.json`**.
3. On the computer on which you're building your device images, copy the default ("neutral") **`oem.json`** file to:

- `%programdata%\Microsoft\Windows\RetailDemo\OfflineContent\OEM\Content\Neutral\HubContent\oem.json`

In addition, create a set for US-English. You must include files for both the default and US-English, regardless of which other languages you support:

- `%programdata%\Microsoft\Windows\RetailDemo\OfflineContent\OEM\US\Content\en-us\HubContent\oem.json`

File paths for localized content (optional):

- `%programdata%\Microsoft\Windows\RetailDemo\OfflineContent\OEM\Content\[locale]\HubContent\oem.json`
- `%programdata%\Microsoft\Windows\RetailDemo\OfflineContent\OEM\[region]\Content\Neutral\HubContent\oem.json`
- `%programdata%\Microsoft\Windows\RetailDemo\OfflineContent\OEM\[region]\Content\[locale]\HubContent\oem.json`

Example: Canada-specific content in French:

- `%programdata%\Microsoft\Windows\RetailDemo\OfflineContent\OEM\CA\Content\fr-ca\HubContent\oem.json`

4. Create a folder of all assets (images and video) and name the folder **`oem_assets`**. Create a zipped version of the **`oem_assets`** folder and name it **`oem.zip`**.
5. Copy the **`oem.zip`** folder of assets for the Retail Demo app to the default and US-English folders:

- `%programdata%\Microsoft\Windows\RetailDemo\OfflineContent\OEM\Content\Neutral\HubContent\oem.zip`
- `%programdata%\Microsoft\Windows\RetailDemo\OfflineContent\OEM\US\Content\en-us\HubContent\oem.zip`

File paths for localized content (optional):

- `%programdata%\Microsoft\Windows\RetailDemo\OfflineContent\OEM\Content\[locale]\HubContent\oem.zip`
- `%programdata%\Microsoft\Windows\RetailDemo\OfflineContent\OEM\[region]\Content\Neutral\HubContent\oem.zip`
- `%programdata%\Microsoft\Windows\RetailDemo\OfflineContent\OEM\[region]\Content\[locale]\HubContent\oem.zip`

6. Build the image.

## Other types of retail demo mode content

In addition to the Retail Demo app and the Attract Loop app, there are other types of content the shopper can see or experience in Retail Demo mode.

### Library content

Many shoppers use the library content (photos, videos, documents) to fully explore the experience within the many apps available on the device or from the Microsoft Store. It is highly recommended to include this content in your image.

### Apps

Include retail demo features in your Windows apps so customers who try out your PCs and devices on the sales floor can jump right in.

We strongly recommend you ensure that the apps included on the device take advantage of the retail demo mode. Not only will your app look amazing for a retail customer, you may even get increased app awareness due to the discovery of your app that leads to post-purchase install.

We recommend the following guidelines when developing retail demo features for your app:

- **Show off, but be focused:** Use the retail mode version of your app as an opportunity to showcase why it rocks and is a reason to buy a Windows device. Put your best foot forward. Keep the story simple: elevator pitch to land your app's value prop.
- **Make sure your app cleans up between uses.**
- **Minimize error and pop-up dialogs:** Error pop-ups invoke a negative experience with the app, Windows and the shopping experience. Minimize pop-ups as much as possible.

To learn more, see [Add retail demo \(RDX\) features to your app](#).

## Add retail demo mode, including language packs, to your images

Add each of the following packages to your images. Note, these packages must be installed in order.

1. If your devices will include multiple languages, add language packs and language interface packs first.

Example:

- Microsoft-Windows-Client-Language-Pack\_x64\_fr-FR.cab
- Microsoft-Windows-Client-Language-Pack\_x64\_vi-VN.cab

Note, do not remove the English language pack, this pack is required for Retail Demo Mode.

2. Next, add the basic language pack for each language, including English. For example:

- Microsoft-Windows-LanguageFeatures-Basic-en-US-Package.cab
- Microsoft-Windows-LanguageFeatures-Basic-fr-FR-Package.cab
- Microsoft-Windows-LanguageFeatures-Basic-vi-VN-Package.cab

3. Next, add the base retail demo pack:

- Microsoft-Windows-RetailDemo-OfflineContent-Content-Package.cab

4. Next, add the localized retail demo pack for each language, including English. Example:

- Microsoft-Windows-RetailDemo-OfflineContent-Content-en-us-Package.cab
- Microsoft-Windows-RetailDemo-OfflineContent-Content-fr-fr-Package.cab
- Microsoft-Windows-RetailDemo-OfflineContent-Content-vi-VN-Package.cab

Available RetailDemo language packs:

- Arabic [ar-SA]
- Bulgarian [bg-BG]
- Chinese
  - Hong Kong SAR (Traditional) [zh-HK]
  - Taiwan (Traditional) [zh-TW]
  - China (Simplified) [zh-CN]
- Croatian [hr-HR]
- Czech [cs-CZ]
- Danish [da-DK]

- Dutch [nl-NL]
- English
  - USA [en-US]
  - UK [en-GB]
- Estonian [et-EE]
- Finnish [fi-FI]
- French
  - Canada [fr-CA]
  - France [fr-FR]
- German [de-DE]
- Greek [el-GR]
- Hebrew [he-IL]
- Hungarian [hu-HU]
- Indonesian [id-ID]
- Italian [it-IT]
- Japanese [ja-JP]
- Korean [ko-KR]
- Latvian [lv-LV]
- Lithuanian [lt-LT]
- Norwegian [nb-NO]
- Polish [pl-PL]
- Portuguese
  - Portugal [pt-PT]
  - Brazil [pt-BR]
- Romanian [ro-RO]
- Russian [ru-RU]
- Serbian (Latin) [sr-Latn-CS]
- Slovak [sk-SK]
- Slovenian [sl-SI]
- Spanish
  - Spain [es-ES]
  - Mexico [es-MX]
- Swedish [sv-SE]
- Turkish [tr-TR]
- Thai [th-TH]
- Ukrainian [uk-UA]
- Vietnamese [vi-VN]

To learn more, see [Add Language Packs to Windows](#).

## Setup and operate retail demo mode

### Resetting the system

In a short time after a shopper stops interacting with the device, the retail demo plays, and Windows begins resetting any sample data in the contacts, photos, and other apps. Depending on the device, this could take between 1-5 minutes to fully reset everything back to normal.

### Retail demo mode capabilities

In Retail Demo mode, shoppers are prevented from modifying key system areas. For example, they won't be able to:

- Add or change the user password
- Change a Microsoft account name
- Access the command line, Registry Editor, or PowerShell utilities
- Anything that requires administrative permissions to the system

## Activate retail mode

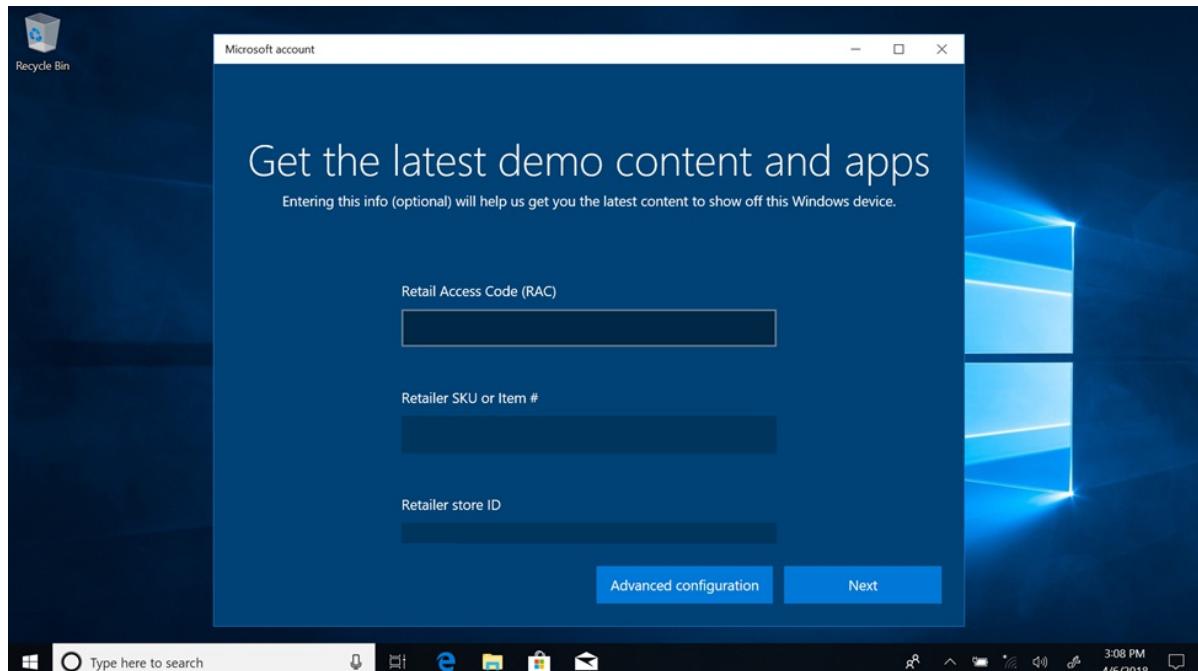
This process can be used to verify that the device is properly configured to launch any custom demo applications. It can also be used to start retail demo mode on a demonstration device.

Use the following process to activate retail demo mode on any retail device.

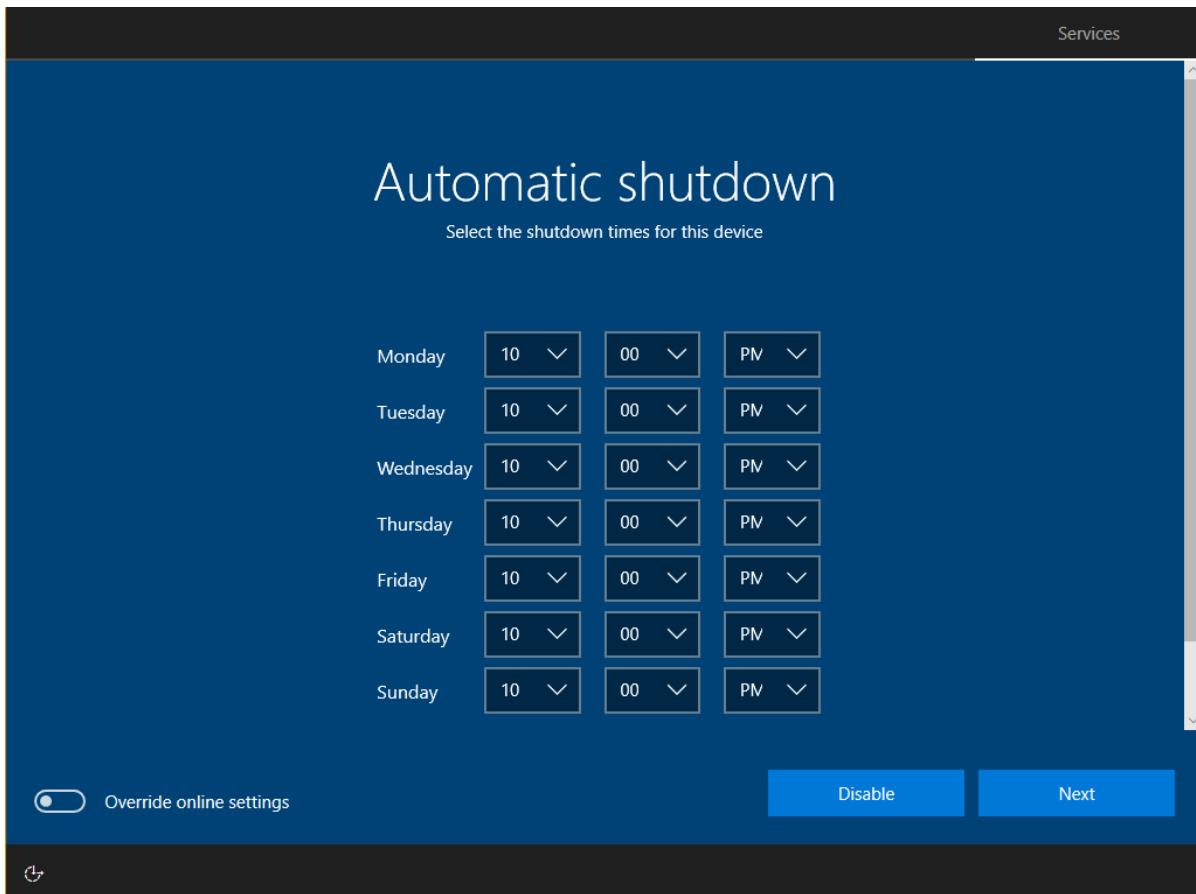
1. Remove the device from the box and press the power button to power up the device.
2. Enter Retail Demo Mode. For instructions, see this section in the [RDX Windows Experience Guide \(WEG\)](#).
3. Proceed with OOBE setup, including acceptance of the legal terms, until you get to RDX setup.
4. In the **Get the latest demo content and apps** page, enter your **Retail Access Code (RAC)**. The available SKUs and items for the RAC are downloaded. Select the **Retailer SKU or Item#**, **Retailer store ID**, and then click **Next**. If the device is offline, go to the next step.

### NOTE

"SKU" is optional and is only required if the retailer associated with the Retail Access Code (RAC) explicitly designed an experience under a specific SKU. The RAC (Retail Access Code), SKU, and Retailer store ID only apply and are only used by Retailers. If this applies to you, the codes should be requested through your Microsoft Account Manager. A list of SKUs will only be available if the retailer associated with the Retail Access Code (RAC) has previously provided it to Microsoft with this intent. If a SKU is not entered, the device will get content specific to the retailer and specific to the model of the hardware.



5. If you would like to specify start up and shut down times, select **Advanced configuration**.
6. In the **Automatic shutdown** page, configure and click **Next**.



7. In the **RDX admin settings** page, you can choose up to 21 days to keep admin account active, or you can choose to keep it active perpetually. A password is required to make an admin account perpetually active.
8. If the device is online, select **Finish** on the next page.

Retail demo setup will reboot the device soon after the desktop appears, and then retail demo configuration will continue in the background until the device begins showing the video attract loop.

### Highlight your device

Retail Demo mode can showcase additional Windows features and apps based on the type of device's form factor.

To access this content, make sure that the [DeviceForm](#) Unattend setting is set.

For example, when **DeviceForm** identifies the device as a Convertible laptop, Retail Demo mode includes content to show off Convertible features in Windows.

### Additional recommendations:

If you are setting up your demo on a tablet device, configure your devices to boot to tablet mode and turn hardware-based prompting off:

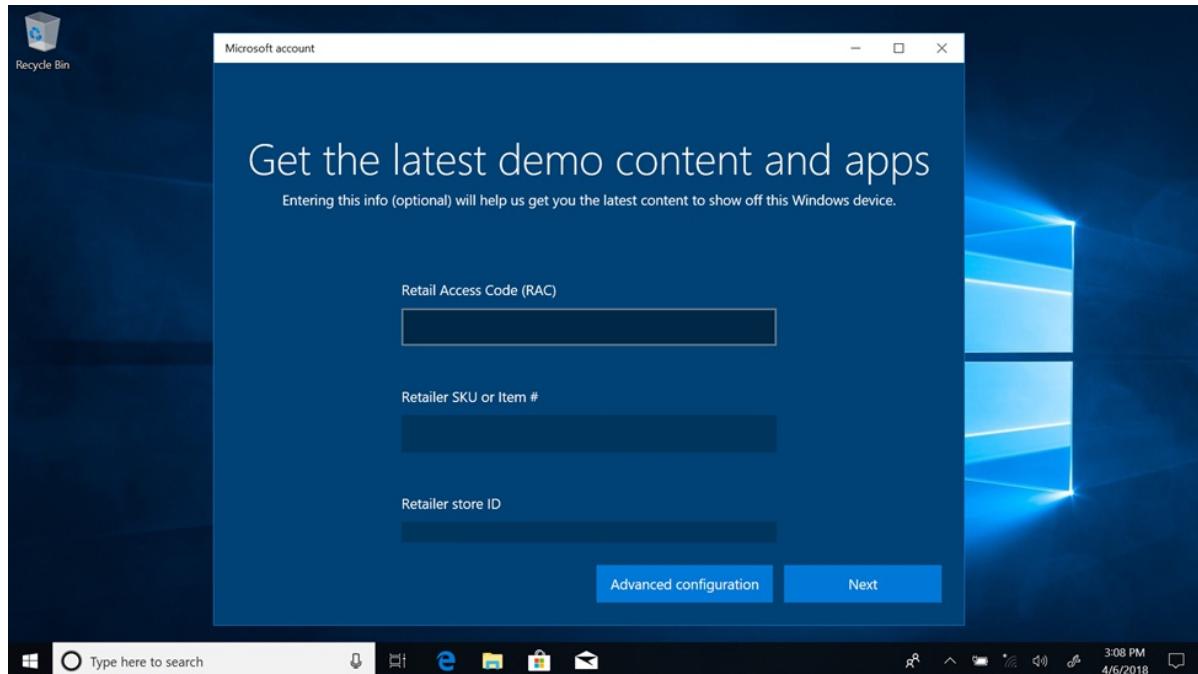
- Make sure that [ConvertibleSlateMode](#) is always accurate for your devices.
- To support booting to tablet mode, configure the [SignInMode](#) setting.
- To remove prompts triggered by changes to [ConvertibleSlateMode](#), configure [ConvertibleSlateModePromptPreference](#) setting.

### Schedule automatic shutdown

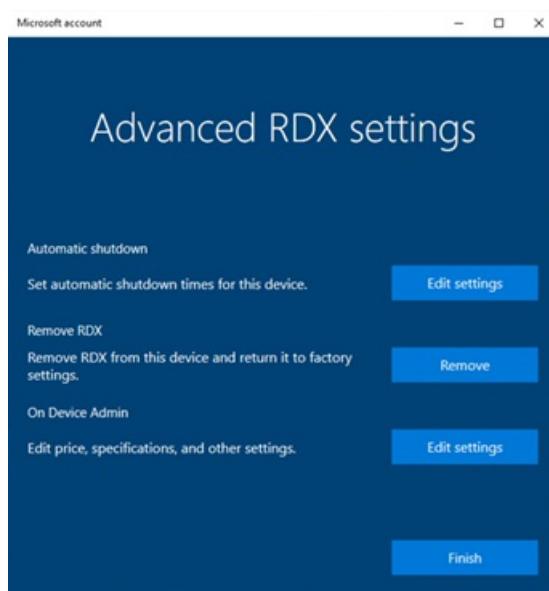
Windows provides a way to set times for the devices to turn on and start retail demo mode automatically. Likewise, you can schedule when the devices shut down. These features allow you to save energy costs, to schedule updates, and to restore the retail demo experience automatically. You can configure shutdown times on the client during the out-of-box experience (OOBE), or post-OOBE during the retail demo mode setup. Additionally, you can configure automatic shutdown times using Retail Demo Asset Management (RDAM) after retail demo mode has been set up.

## To schedule shutdowns on a local device

1. Open retail demo mode configuration. For instructions, see this section in the [RDX Windows Experience Guide \(WEG\)](#).
2. The retail demo mode configuration UI is displayed. Select the **Advanced configuration** button.



3. In the Advanced RDX settings page, under **Automatic Shutdown**, select **Edit settings**. This allows you to configure the automatic shutdown of the device.



## Remove retail demo components

After a customer completes the out-of-box experience (OOBE), Windows schedules the removal of all RDX components, including any customizations you've added in the `%programdata%\Microsoft\Windows\RetailDemo\OfflineContent\` folder.

For devices with more than 32GB of storage, the components are automatically removed 7 days after the customer completes OOBE.

For devices with 32GB of storage or less, by default, the components are automatically removed 30 minutes after the customer completes OOBE. To change this schedule, find the registry key

`HKEY\Software\Microsoft\Windows\CurrentVersion\Setup\OOBE` and set the value `DeleteDemoContentDelay` to a number

of minutes from 30 and 10080 (= 7 days).

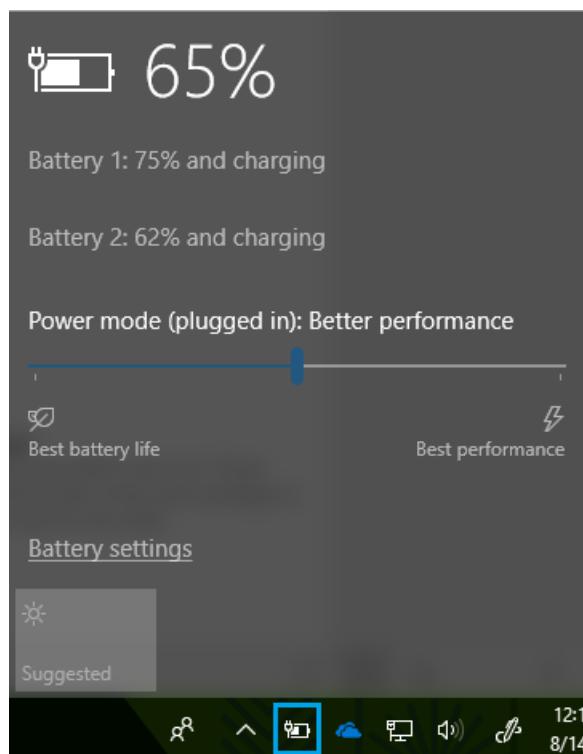
```
md c:\HWID
Set-Location c:\HWID
Set-ExecutionPolicy Unrestricted
Install-Script -Name Get-WindowsAutoPilotInfo
Get-WindowsAutoPilotInfo.ps1 -OutputFile AutoPilotHWID.csv
```

# Customize the Windows performance power slider

10/2/2018 • 8 minutes to read • [Edit Online](#)

The Windows performance power slider enables end customers to quickly and intelligently trade performance of their system for longer battery life. As a customer switches between the four slider modes to trade performance for battery life (or vice versa), Windows power settings are engaged behind the scenes. You are able to customize the default slider mode for both AC and DC, and can also configure the power settings, and PPM options, that are engaged for each slider mode.

Customers can access the slider on their Windows device by clicking or tapping the battery icon in the task bar. The slider appears in the battery flyout.



Customers can choose their power mode by moving the slider to the left and right. Customers can choose to prioritize the remaining battery life on the device, or the performance of apps and services running on the device. The screenshot above shows the slider is in the **Better Performance** slider mode, which is the out-of-box Windows default.

## Slider availability

The Windows power slider is available for AMD and Intel platforms running Windows 10, build 1709 and newer builds of Windows. It is not available on devices with ARM64 processors. The slider will appear on a device only when the **Balanced** power plan, or any plan that is derived from Balanced, is selected. There is not an option for either users or OEMs to remove the slider UX.

Devices that have the High Performance, Power Saver, or any "OEM Recommended" power plans will not be disturbed during the Windows upgrade process. If a user upgrades from a version of Windows that does not support the slider, to a version that does, there will be no change to their High Performance, Power Saver, or "OEM Recommended" power plan. These users will not see the slider UX, and they can still configure their power plans in the same way they could before upgrading.

Users will see the power slider appear only when they apply the Balanced power plan from the **Settings** app,

under **System > Power & Sleep > Additional power settings**.

#### NOTE

After the user changes to a Balanced performance plan, there is no way for them to go back to using the High Performance plan from the UI, although it is possible from the cmd line (via powercfg).

## Guidance for High Performance devices

If you ship a device with a High Performance power plan, such as a gaming device, consider applying the same settings that are defined on the High Performance plan to the Balanced power plan. For example, if the timeout value for powering off the HDD or Display is set to X or Y on High Performance, apply those same values on Balanced.

You can also customize power settings for each of the slider modes in your firmware. See [Configure power settings and PPM options](#) for more information.

## Set default power slider mode

Customers can choose one of four slider modes:

- **Battery Saver:** Helps conserve power, and prolong battery life, when the system is not connected to a power source. When battery saver is on, some Windows features are disabled, throttled, or behave differently. Screen brightness is also reduced. Battery Saver is only available on DC. To learn more, see [Battery Saver](#).
- **Better Battery:** Delivers longer battery life than the default settings on previous versions of Windows. Available on both AC and DC. In some cases, users will see this mode labeled **Recommended**, rather than **Better Battery**, in their slider UI.
- **Better Performance:** Default slider mode that slightly favors performance over battery life and is appropriate for users who want to tradeoff power for better performance of their apps. Available on both AC and DC.
- **Best Performance:** Favors performance over power and is targeted at users who want to tradeoff power for performance and responsiveness. Available on both AC and DC.

#### NOTE

[Game mode](#) operates independently of the Windows performance power slider, and can be engaged in any slider mode.

## To set the default slider mode

You can configure the default slider mode for both AC and DC. If a customer chooses a different slider mode on either AC or DC, their selection will become the new default setting.

#### NOTE

Battery Saver is not available as a default slider mode.

First, create a provisioning package using [Windows Configuration Designer](#). You will then edit the customizations.xml file contained in the package to include your power settings. Use the XML file as one of the inputs to the Windows Configuration Designer command-line to generate a provisioning package that contains the power settings, then apply the package to the image. For information on how to use the Windows Configuration Designer CLI, see [Use the Windows Configuration Designer command-line interface](#).

WINDOWS PROVISIONING PATH	PROVISIONING SETTING NAME	VALUES
Common\Power\Controls\Settings\ {setting name}	<b>DefaultOverlayAcPowerScheme:</b> Default slider mode for AC <b>DefaultOverlayDcPowerScheme:</b> Default slider mode for DC	<b>BetterBatteryLife:</b> Sets the slider to the Better Battery mode <b>MaxPerformance:</b> Sets the slider to the Best Performance mode

#### NOTE

If no default is configured, Better Performance will be the default slider mode for both AC and DC.

#### XML Example

Below is an example customizations.xml file that defines default slider modes.

```
<?xml version="1.0" encoding="utf-8"?>
<WindowsCustomizations>
  <PackageConfig xmlns="urn:schemas-Microsoft-com:Windows-ICD-Package-Config.v1.0">
    <ID>{7e5c6cb3-bd16-4c1a-aacb-98c9151d5f20}</ID> <!-- ID needs to be unique GUID for the package -->
    <Name>CustomOEM.Power.Settings.Control</Name>
    <Version>1.0</Version>
    <OwnerType>OEM</OwnerType>
  </PackageConfig>
  <Settings xmlns="urn:schemas-microsoft-com:windows-provisioning">
    <Customizations>
      <Common>
        <Power>
          <Controls>
            <DefaultOverlayDcPowerScheme>"BetterBatteryLife"</DefaultOverlayDcPowerScheme>
            <DefaultOverlayAcPowerScheme>"MaxPerformance"</DefaultOverlayAcPowerScheme>
          </Controls>
        </Power>
      </Common>
    </Customizations>
  </Settings>
</WindowsCustomizations>
```

## Configure power settings and PPM options engaged by the slider

You can use overlays to customize the power settings and PPM options that are engaged for each slider mode. In previous versions of Windows, power settings could only be configured per power scheme, and PPM options could only be configured per power profile. The introduction of overlays enables OEMs to better optimize power settings based on the slider mode selected by the user, as opposed to the power scheme or power profile selected by the device.

To configure PPM and power settings per slider mode, apply them to one of the following overlays:

- **BetterBatteryLifeOverlay**
- **MaxPerformanceOverlay**

The Battery Saver mode inherits the settings configured for the Constrained PPM profile. The Best Performance mode inherits the settings configured for the Balanced (default) profile. Configure these profiles to customize the settings that are engaged in the associated slider modes.

#### **NOTE**

Settings such as disk and display timeouts, and other legacy power settings, are not customizable via the performance/power slider. Only settings which can affect perceived performance differences can be customized across slider modes. Each slider mode should be thought of as a “lite” power plan, which only contains settings that impact performance, such as CPU settings (PPM) and power throttling. Other factors which control performance (GPU, thermals etc) are in OEM/SVs control and they can create custom power-settings for those and connect them to the slider via the INF.

### **Configure PPM optimization**

Optimizing PPM enables the OS to favor either power or performance, depending on user preference (similar to the low power media profile that is applied when a user is watching video in full screen mode). PPM settings should favor battery life for the Battery Saver and Better Battery slider modes, and favor performance for the Better and Best Performance slider modes.

PPM options can be configured for all AMD and Intel platforms using Windows Provisioning Framework. To learn more about the PPM options that you can configure, and how to configure them per power scheme, see Processor power management options.

#### **XML Example**

Below is an example of a customizations.xml file that uses overlays to define PPM settings for the Better Battery and Best Performance slider modes.

```
<Power>
  <Policy>
    <Settings>
      <Processor>
        <SchemePersonality>
          <!-- EPP override for default PPM profile for "Better Battery" -->
          <Profile SchemeAlias=" BetterBatteryLifeOverlay">
            <Setting ProfileAlias="Default">
              <PerfEnergyPreference>
                <DcValue>60</DcValue>
              </PerfEnergyPreference>
            </Setting>
          </Profile>
          <!--EPP override for default PPM profile for "Best Performance" -->
          <Profile SchemeAlias="MaxPerformanceOverlay">
            <Setting ProfileAlias="Default">
              <PerfEnergyPreference>
                <DcValue>30</DcValue>
              </PerfEnergyPreference>
            </Setting>
          </Profile>
        </SchemePersonality>
      </Processor>
    </Settings>
  </Policy>
</Power>
```

### **Configure performance and power settings**

To engage your customized power settings only when the slider is in a particular mode, create an **AddPowerSettingDirective** in your INF file that indicates the default values for each overlay. There are **Default** directives that must be included in an **AddPowerSetting** section. A **Default** directive specifies the three overlays that apply to an AC and DC power state each.

Add the following three directives to define settings for the various slider modes:

SLIDER MODE	INF GUID	PPKG SCHEMEALIAS
Better Battery	{961CC777-2547-4F9D-8174-7D86181b8A7A}	BetterBatteryLifeOverlay
Better Performance	{381B4222-F694-41F0-9685-FF5BB260DF2E}	Balanced
Best Performance	{DED574B5-45A0-4F42-8737-46345C09C238}	MaxPerformanceOverlay

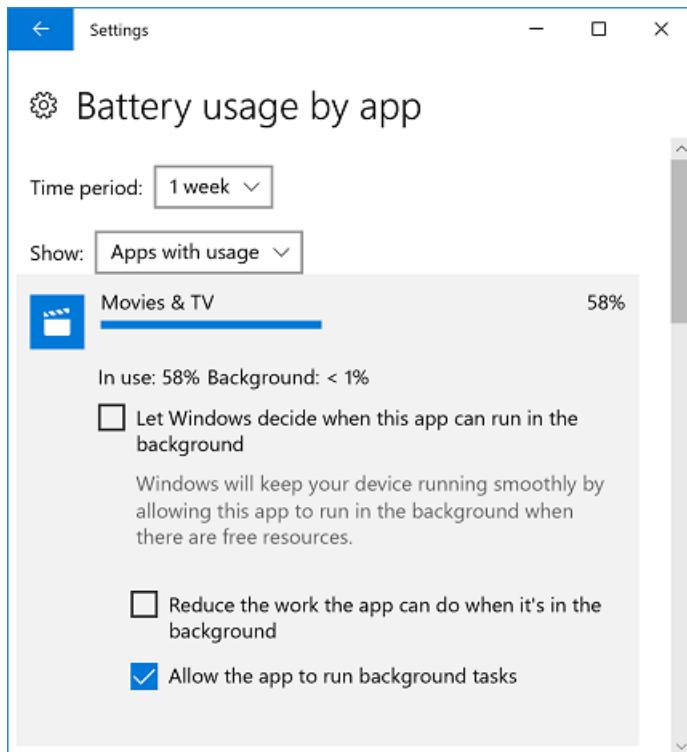
See [INF AddPowerSetting Directive](#) for further instructions.

## Power throttling

Most Windows users have multiple apps running on the operating system at the same time, and often, the apps running in the background consume significant power. Windows leverages modern silicon capabilities to run background work in an energy-efficient manner, significantly enhancing battery life. Power throttling saves up to 11% in CPU power by throttling CPU frequency of applications running in the background. With power throttling, when background work is running, Windows places the CPU in its most efficient operating modes. Learn more about this feature in our blog post: [Introducing power throttling](#).

Power throttling does not suspend or close apps and services on the device.

Power throttling is always engaged, unless the slider is set to **Best Performance**. In this case, all applications will be opted out of power throttling. Users can also opt individual apps out of power throttling in the Battery usage UX:



OEMs do not have an option to disable or change power throttling on any of the Windows slider modes.

### NOTE

Power throttling is available for devices using Intel's 6th or 7th generation processors (including those without Intel's SpeedShift technology) only.

# Query for power slider settings

There are two logs you can utilize to query the performance power slider settings defined on an OS image:  
Powercfg output, and Event Tracing for Windows (ETW) logs.

## PowerCfg output

Run `"powercfg /qh > output.txt"` from an [elevated command prompt](#), then open output.txt in any text editor to view the settings.

## Event tracing for Windows (ETW) logs

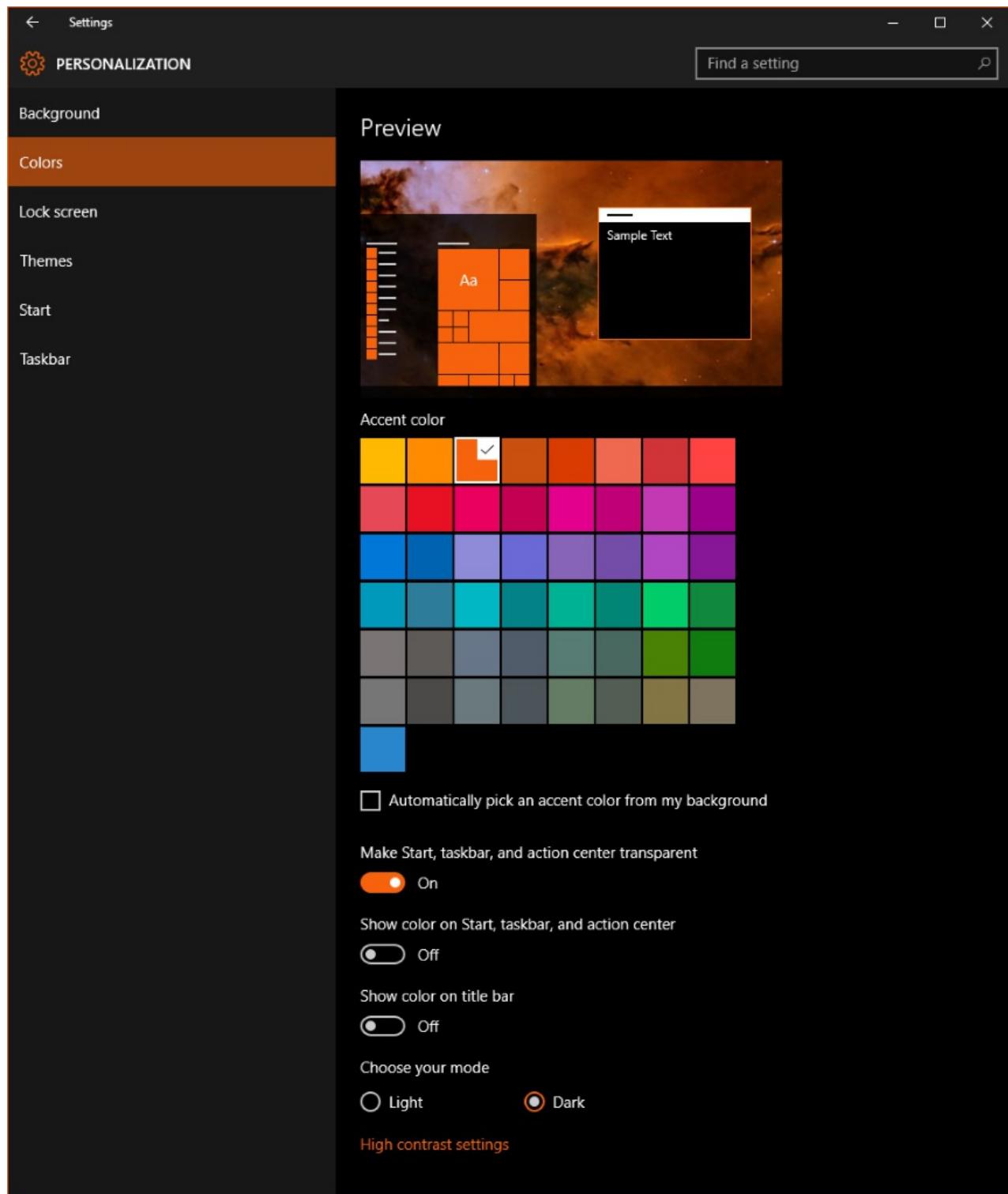
Use the inbox WPRUI.exe or WPR.exe to collect an ETW log with the POWER scenario enabled. To collect and analyze the ETW log:

1. Launch an elevated command prompt window
2. Enter the command: `WPR -start power -filemode`
3. Using the power slider UX, move the slider to each of the four modes
4. Go back to the elevated command prompt window and enter the command:  
`WPR -stop PerfPowerSliderSettings.etl`
5. Open **PerfPowerSliderSettings.etl** in the [Windows Performance Analyzer](#) (WPA) tool. WPA comes bundled with the [Windows Assessment and Deployment Kit](#) (Windows ADK).
6. Click on **Trace**.
7. Click on **System Properties** then **System configuration**.
8. In the new tab that opens, click on **Power Settings**.

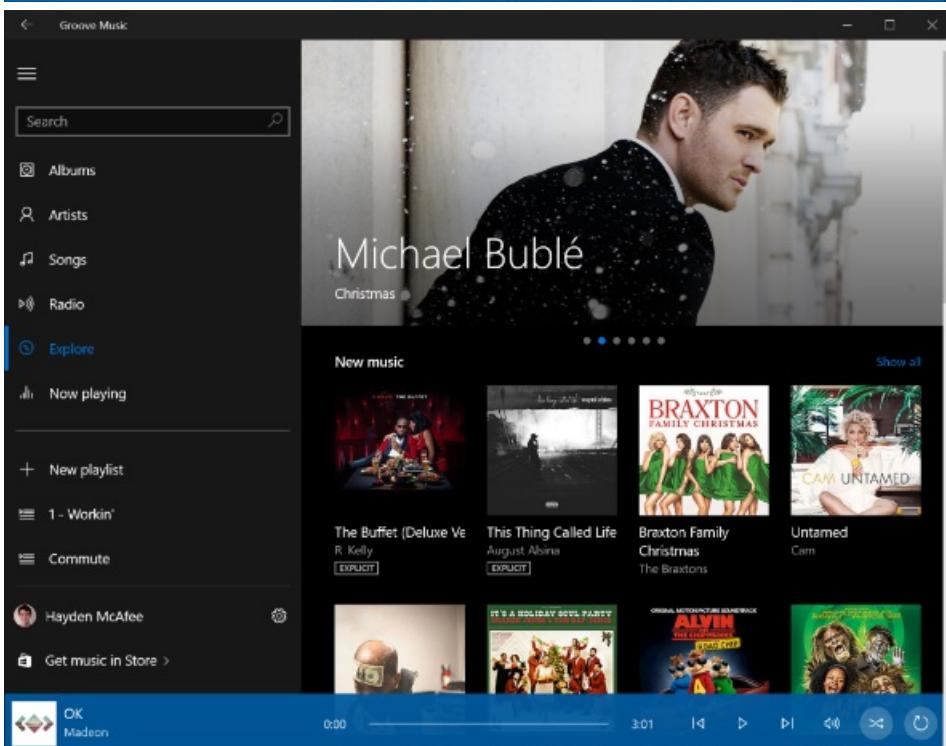
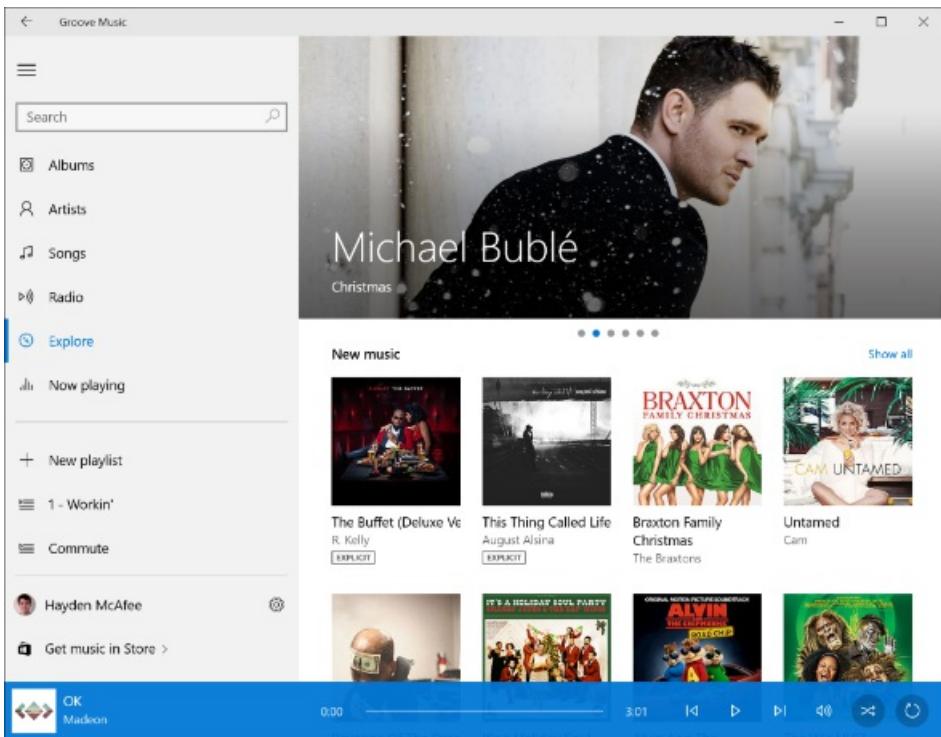
# Set dark mode

10/2/2018 • 2 minutes to read • [Edit Online](#)

This personalization setting for end users allows them to express preference whether to see applications which support the setting in a dark or light mode.



Many Microsoft first party applications apply the setting and it is easy for you to support the functionality in your UWP applications as well.



You can customize the default Windows theme via Unattend.xml. The Unattend component includes a setting `UWPAppsUseLightTheme` that configures dark mode as the default for apps that support it.

```
<settings pass="oobeSystem">
    <Themes>
        <ThemeName>MyOLEDTheme</ThemeName>
        <DefaultThemesOff>false</DefaultThemesOff>
        <DesktopBackground>c:\windows\OLEDFriendlyImage.jpg</DesktopBackground>
        <WindowColor>Lime</WindowColor>
        <UWPAppsUseLightTheme>false</UWPAppsUseLightTheme>
    </Themes>
</settings>
```

To learn more about customizing Windows themes, see [Themes](#) in the Unattended Windows Setup Reference.

# Customize the Get Help app

10/2/2018 • 4 minutes to read • [Edit Online](#)

The Get Help app empowers customers to self-help with troubleshooters, instant answers, Microsoft support articles, and more, before contacting assisted support.

If you have a support app or support website you would like to direct customers towards, you can use unattend.xml to display your support option within the Get Help app. A link to your support app or website is surfaced wherever options to contact support are shown in the Get Help app. The first item in the list will be the link you provided.

Customers are sent to the Get Help from the Settings app, Cortana, Bing Instant Answers, the Start Menu, and numerous Microsoft web experiences. It is also possible to launch Get Help from your own apps and websites.

## Consumer experience

For consumers, the Get Help app provides a way to ask a question and get recommended solutions or contact assisted support. Depending on the country/region and language of the device, one of two experiences will be shown: Virtual Agent, or Search support.

### **Virtual Agent**

Microsoft's Virtual Agent is a support chat bot designed to help with issues related to Windows and other products. This brings a conversational approach to understanding problems and providing the most appropriate solution. If the Virtual Agent is unable to provide a solution, it will direct customers to the options for contacting support; it is also possible to ask for those options at any time. This experience is only available in en-US.

OEM customization provides the top support option in the list — a link to either your support website, or your support app.

[← Get Help](#)

 Virtual agent [Start over](#)



Hello John!

I'm Microsoft's new virtual support agent. Describe your problem and I'll look for the best solution.

how do i change printers



Enter your response



## Search support

In markets that do not have the Virtual Agent experience available, customers can utilize search support by entering a question and receiving back recommended support content. Beneath the search results, the options for contacting support will be listed.

OEM customization provides the top support option in the list — a link to either your support website, or your support app.



## How can we help you?

Tell us about the **problem** that you're having  
and which **product** you're using

I am having sound issues on my usb speakers since I have upgraded to windows 10

Next

### Disability Answer Desk

Support for people with disabilities.

### Try this help content

#### No sound in Windows - Windows Help

[windows.microsoft.com/en-us/windows/no-sound-help](https://windows.microsoft.com/en-us/windows/no-sound-help)

you identify and fix common **sound** problems in **Windows**, including no **sound** coming from your speakers or some problem...

#### Tips for fixing common **sound** problems - Windows Help

[windows.microsoft.com/en-us/windows/tips-fixing-common...](https://windows.microsoft.com/en-us/windows/tips-fixing-common...)

Step-by-step tutorial to help you identify and fix common sound problems, go to No **sound** in **Windows**. Show sound from my...

#### Tips for fixing common **sound** problems - Windows Help

[windows.microsoft.com/en-us/windows/tips-fixing-common...](https://windows.microsoft.com/en-us/windows/tips-fixing-common...)

Step-by-step tutorial to help you identify and fix common sound problems, go to No **sound** in **Windows**. Show sound from my...



### Get more support

#### Contact Contoso

If your device (ModelXYZ) is still under warranty, contact Contoso for support.

#### Chat

There are **12 users** waiting.

## Enterprise experience

For Enterprise SKUs, the Get Help app provides a different experience that focuses on getting customers to the right kind of support. The support options listed are shown to all enterprise customers. Availability of support within each option depends upon support agreements with the enterprise.

### NOTE

OEM support options are not displayed in the Enterprise experience of the Get Help app.

## Customize support information

To display your OEM support information in the Get Help app, you must provide either a link to the URL of your support website, or to the URI of your support app, in Unattend.xml under

`Microsoft-Windows-Shell-Setup-OEMInformation`.

See the [OEMInformation setting](#) in the Unattended setup reference to learn more about how to add your support information to the Get Help app.

### Link to your support app

Here is an example where a path for `SupportAppURL` is supplied. In this case , the Get Help app will direct customers to the OEM's support app:

```
<OEMInformation>
  <SupportProvider>Contoso</SupportProvider>
  <SupportAppURL>ContosoSupport://path/?param=val</SupportAppURL>
</OEMInformation>
```

`SupportAppURL` must be present and contain valid string values, otherwise Get Help won't pick up your support information. `SupportProvider` is an optional override for the name shown on the link; the default when `SupportProvider` is not present is `SystemManufacturer` from `SystemInformation` (`msinfo32.exe`).

"ContosoSupport" is a sample protocol name; you can pick your app's own protocol name, if it does not conflict with an existing protocol name in the system.

To register a protocol handler for your app:

- For a Universal app, the protocol handler is specified in the package.appxmanifest file (part of the Visual Studio project), under the `<Extensions>` section. See [Handle URI activation](#) for more details.
- For a Win32 app, the protocol handler is specified in the registry. See [Registering an Application to a URI Scheme](#) for more details.

### NOTE

Win32 apps are not supported in Windows 10 in S mode.

### Link to your support website

Here is an example where a URL for `SupportURL` is provided. In this case, the Get Help app will direct customers to the OEM's support webiste.

```
<OEMInformation>
  <SupportProvider>Contoso</SupportProvider>
  <SupportURL>https://www.contoso.com/support?param=val</SupportURL>
</OEMInformation>
```

`SupportURL` must be present and contain valid string values, otherwise Get Help won't pick up your support information. `SupportProvider` is an optional override for the name shown on the link; the default when `SupportProvider` is not present is `SystemManufacturer` from `SystemInformation` (`msinfo32.exe`).

The Get Help app will launch the specified `SupportURL` in Microsoft Edge when the OEM support option is chosen.

## Launch Get Help

You can send customers to the Get Help app from your app or website by providing a link to the following URL:

```
ms-contact-support://oem/<Manufacturer>
```

Where `<Manufacturer>` is an all lowercase, unbroken name such as "contoso" or "fabrikaminc". Generally, this should be the simplest version of your brand name, not the longer formal business name. This information is used to identify where users launched the Get Help app from; it is not used to customize the app directly.

# Customize SIM card slot names

10/2/2018 • 2 minutes to read • [Edit Online](#)

In Windows 10, version 1803, you can customize the names of SIM card slots on the device to more easily differentiate between them. For example, if the device has both an embedded SIM slot and an external SIM slot, customizing the names will help your customers understand which is which.

## IMPORTANT

Only devices with a Dual SIM Single Activation (DSSA) configuration support this customization.

The SIM card slot names that you choose are displayed in **Settings**, under **Network & Internet > Cellular**. If no custom names are provided, the default names are **SIM1** and **SIM2**.

## Instructions

1. Create a provisioning XML file (prov.xml). For instructions, see [Create a Prov.xml](#).
2. Add the following XML to your provisioning XML file:

```
<wap-provisioningdoc>
    <characteristic type="Registry">
        <characteristic type="HKLM\Software\Microsoft\Cellular\MVSettings\DeviceSpecific\CellUX">
            <parm name="SlotSelectionSim1Name" value="Your SIM name 1" datatype="string"/>
            <parm name="SlotSelectionSim1Name" value="Your SIM name 2" datatype="string"/>
        </characteristic>
    </characteristic>
</wap-provisioningdoc>
```

3. Replace "Your SIM name 1" and "Your SIM name 2" with the desired names for your SIM card slots.
4. Create a resource-only .dll for the localized versions of your SIM card slot names. See [Create a resource-only .dll for localized strings](#) for instructions.
5. In your resource-only .dll, set the `BaseD11` asset to point to the location of your base MUI DLL file. For example: `C:\Path\DisplayStrings.dll`.
6. Add the language MUI packages (\*.dll.mui) for all the languages you are supporting and have localized strings for. To do this:
  - Set the asset's `Name` to `LanguageD11/$(langid)` where `$(langid)` corresponds to the language. For example: `LanguageD11/en-US`.
  - Set the asset's `Source` to the location of the .dll.mui file for that language. For example: `C:\Path\en-us\DisplayStrings.dll.mui`.
  - Repeat these steps for other languages. For example, the following XML has entries for en-US, fr-CA, and es-MX languages.

```
<Asset Name="LanguageD11/en-US" Source="C:\Path\en-us\DisplayStrings.dll.mui" />
<Asset Name="LanguageD11/fr-CA" Source="C:\Path\fr-CA\DisplayStrings.dll.mui" />
<Asset Name="LanguageD11/es-MX" Source="C:\Path\es-MX\DisplayStrings.dll.mui" />
```

## Related topics

Create a resource-only .dll for localized strings

Customizations for mobile devices

# Customize a Specific Absorption Rate (SAR) mapping table

10/2/2018 • 5 minutes to read • [Edit Online](#)

You can configure and store a Specific Absorption Rate (SAR) table for mobile broadband modems in the registry. When a mobile broadband modem is connected to the Windows device, Windows automatically uses the table to map the mobile country code (MCC) of the modem's registered mobile operator (MO) to its appropriate SAR back-off index, and configure the modem with it.

You may choose to configure the registry settings at imaging time, or run-time. If you build the registry settings into the image at image deployment time within a package, the SAR mapping table will be ready for any OS component as soon as it starts. If you use a run-time component to configure the registry settings after device bootup, you ensure that the static SAR configuration will not be changed and/or wiped out by Windows installation or upgrade, and that it stays consistent to the device and independent of OS installation.

For more details on SAR support for mobile broadband modems, please see [Mobile Broadband Specific Absorption Rate Platform Support](#).

Here is an overview of how Windows will read and configure the modem based on your customized SAR mapping table:

1. Create a package that contains your registry settings, including those for the [SARMappingTable](#) and [SARConfiguration](#).
2. Build the package into the image for the device.
3. Windows (the WWAN service, in particular) will read the registry at start-up and store the settings for later usage when an embedded, SAR-capable modem registers with a particular MO.
4. Windows also listens to registry change notifications to know if the registry for the settings is changed. This means you may use your own way of adding and changing the settings at run-time, and Windows will accept the changes immediately.
5. When a modem is registered with an MO at run-time, Windows takes the MCC of the MO and finds the corresponding SAR back-off index(es) from the SAR mapping table.
6. Windows will then send the SAR back-off index to the modem using the MBIM interface defined in [Mobile Broadband Specific Absorption Rate Platform Support](#).
7. When the modem roams to another country, the MCC for the new MO will change. Windows will again find the corresponding SAR back-off index(es) from the SAR mapping table using the MCC of the new MO and send it to modem.

## Registry location and syntax

The registry settings to build and configure the SAR mapping table reside exclusively under the base registry key:

`HKLM\OEM\Cellular\DeviceSpecific`

Under the base key, there are two subkeys:

- [SARMappingTable](#): contains the SAR back-off index mapping table.
- [SARConfiguration](#): contains control settings.

Setting these subkeys is entirely optional. You may supply static SAR configuration settings at image-time or update any static settings at run-time.

#### NOTE

If you have components update the settings at run-time, you must increment the configuration version number in the registry value `ConfigurationVersion` as the last write to the registry. Whenever the `ConfigurationVersion` registry value is changed, Windows will read all configuration settings and put them into effect.

## SARMappingTable subkey

The `SARMappingTable` subkey may have up to 1000 registry values. The SAR back-off index(es) is per country. The SAR back-off table will be able to support one entry per country. A country in this context is identified by the standard MCC (Mobile Country Code).

#### NOTE

The value name must consist of three decimal-digit characters that represents the MCC. There may be up to 1000 registry value names, "000" through "999".

VALUE NAME	TYPE	DATA
<i>Three-decimal-digit representing the MCC</i>	WCHAR string	Comma-separated decimal number in WCHAR string, such as <code>0,2,5,8</code> . The numbers represent the SAR back-off indexes for the MCC. The sequence of back-off indexes corresponds to an array of antennas in modem, with the first back-off index for the antenna at index 0, the second back-off index for the antenna at index 1, and so on. For a simple modem with only one antenna, there needs to be only one index in the string, such as "2", for the first and only antenna.

If a registry value for a particular MCC is absent, the data in the special reg value `000` will be used. You may use this default value for countries that do not need specific back-off indexes. If both a registry value for the MCC and the special reg value `000` are absent, no SAR index will be used for the MCC.

## SARConfiguration subkey

The `SARConfiguration` settings do not affect your ability to use modem DSI messages to pass through. For example, SAR proxy may implement a custom design for SAR control and mapping using the existing API (the WWAN service API and/or the corresponding WinRT APIs).

For the `BackOffEnabled` and `ControlMode` settings, the value in modem DSI messages will take precedence. If a modem DSI message passes through the WWAN service, the values of these two settings will be saved and will be used next time they are needed, regardless what values the registry settings for those are. If the `BackOffEnabled` and `ControlMode` settings in registry contain `0xFFFFFFFF` (no change) and no modem DSI message ever passes through, the WWAN service will use the value currently in the modem. The WWAN service queries the modem at start to obtain and remember the values in the modem.

Value Name	Type	Data
SARMappingTableEnabled	DWORD	<p><input type="checkbox"/> 0 - SAR mapping table is disabled.</p> <p><input checked="" type="checkbox"/> 1 - SAR mapping table is enabled.</p> <p>If the data is absent or invalid, the default value of <input type="checkbox"/> 0 is applied.</p>
BackoffEnabled	DWORD	<p><input type="checkbox"/> 0 - SAR back-off is disabled</p> <p><input checked="" type="checkbox"/> 1 - SAR-backoff is enabled.</p> <p><input type="checkbox"/> 0xFFFFFFFF – modem should retain its current back-off state.</p> <p>If the data is absent or invalid, the default value of <input type="checkbox"/> 0xFFFFFFFF is applied.</p>
ControlMode	DWORD	<p><input type="checkbox"/> 0 - SAR back-off mechanism is controlled by the modem device directly.</p> <p><input checked="" type="checkbox"/> 1 - SAR-backoff mechanism is controlled and managed by the operating system.</p> <p><input type="checkbox"/> 0xFFFFFFFF – modem should retain its current control mode.</p> <p>If the data is absent or invalid, the default value of <input type="checkbox"/> 0xFFFFFFFF is applied.</p>
ConfigurationVersion	DWORD	<p>This registry value is designed for OEM run-time components to inform Windows that the SAR mapping table and other parameters are updated. An OEM run-time component must increment the <input type="checkbox"/> ConfigurationVersion registry value every time it completes updating the SAR mapping table, or other parameters in the registry.</p> <p>If the data is absent or invalid, the run-time component will not configure any SAR registry settings.</p>

# Customize Windows Pen and Ink

10/8/2018 • 2 minutes to read • [Edit Online](#)

Customers see the the **Pen & Windows Ink** workspace when they click **Settings > Devices > Pen and Windows Ink**.

Windows provides a few means for you to customize the Pen and Ink workspace:

- You can create an advanced Pen settings app, and link to it directly within the Pen & Windows Ink Settings using Unattend.xml. See [Microsoft-Windows-TwinUI | CustomProtocol](#).
- You can hide the Pen shortcut settings from the Pen & Windows Ink Settings using Unattend. This is helpful for devices that are not compatible with pen settings. See [Microsoft-Windows-TwinUI | Hide](#).
- You can add up to three links to your own apps to the Pen & Windows Ink settings.
- You can hide the “Ignore touch input when I’m using my pen” option within Pen & Windows Ink settings.

## Pin up to three apps in Pen & Windows Ink Settings

Starting in Windows 10, build 1703, you can pin up to three UWP apps in the frequently used apps section of the Windows Ink Workspace. You do this by creating an xml file called `InkWorkspaceModification.xml` and placing it in the following directory: `%SystemDrive%\Users\Default\AppData\Local\Microsoft\Windows\Shell`.

In the `InkWorkspaceModification.xml` file, you link to UWP apps by creating a `Tile` element and providing the `AppUserModelID`. If you are using desktop apps instead, you'd add a `DesktopApplicationTile` element and provide the `LinkFilePath`.

```
<?xml version="1.0" encoding="utf-8"?>
<LayoutModificationTemplate
    xmlns="http://schemas.microsoft.com/Start/2014/LayoutModification"
    xmlns:defaultlayout="http://schemas.microsoft.com/Start/2014/FullDefaultLayout"
    xmlns:start="http://schemas.microsoft.com/Start/2014/StartLayout"
    xmlns:taskbar="http://schemas.microsoft.com/Start/2014/TaskbarLayout"
    Version="1">
    <InkWorkspaceTopApps>
        <Tile AppUserModelID="Microsoft.WindowsFeedbackHub_8wekyb3d8bbwe!App"/>
        <Tile AppUserModelID="Microsoft.WindowsCalculator_8wekyb3d8bbwe!App"/>
        <DesktopApplicationTile LinkFilePath="%APPDATA%\Microsoft\Windows\Start Menu\Programs\OneDrive.lnk"/>
    </InkWorkspaceTopApps>
</LayoutModificationTemplate>
```

## Hide the “Ignore touch input when I’m using my pen” setting

In Windows 10, build 1703, you have the option to hide the **Ignore touch input when I’m using my pen** setting if the device doesn’t accept touch and pen input simultaneously.

 Home

## Pen &amp; Windows Ink

Find a setting 

Devices

 Printers & scanners Connected devices Bluetooth Mouse Touchpad Typing Pen & Windows Ink AutoPlay USB

Choose which hand you write with

 Right Hand ▾

Show Visual Effects

 On

Show cursor

 On

Ignore touch input when I'm using my pen

 Off

Show the handwriting panel when not in tablet mode and there's no keyboard attached

 On

Improve how well your PC recognizes your handwriting. Try it out with words your PC sometimes misunderstands. Not all languages are supported.

[Get to know my handwriting](#)

To hide the simultaneous pen and touch settings UI, create the following registry key as a DWORD and set it to a 1. Setting it to 0 (default) will show the setting again.

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Pen\HideSPTSettings
```

# Customizations for enterprise desktop

10/2/2018 • 2 minutes to read • [Edit Online](#)

Windows 10 Enterprise customizations provide a controlled and specialized experience for the end-users of a Windows 10 device by allowing OEMs and system administrators to lock down the Windows 10 device interaction experience.

There are many reasons for locking down a device, such as protecting the system from malicious users, providing a custom defined user experience, and increasing system reliability.

You can lock down your Windows 10 desktop device by using the lock down features individually or in combination with each other to get the effect you desire for your image. You can, for instance, create a dedicated cashier device that runs a full screen Point of Service (POS) application.

TOPIC	DESCRIPTION
<a href="#">Custom Logon</a>	You can use the Custom Logon feature to suppress Windows 10 UI elements that relate to the Welcome screen and shutdown screen. For example, you can <a href="#">suppress all elements of the Welcome screen UI</a> and provide a custom logon UI. You can suppress the ease of access option on the logon screen. You can also suppress the Blocked Shutdown Resolver (BSDR) screen and automatically end applications while the OS waits for applications to close before a shutdown.
<a href="#">Keyboard Filter</a>	Use Keyboard Filter to suppress undesirable key presses or key combinations. Normally, a customer can use certain Windows key combinations like <b>Ctrl+Alt+Delete</b> or <b>Ctrl+Shift+Tab</b> to alter the operation of a device by locking the screen or using Task Manager to close a running application.
<a href="#">Shell Launcher</a>	Use Shell Launcher to replace the default Windows 10 shell with a custom shell. You can use almost any application or executable as your custom shell, such as a command window or a custom dedicated application.
<a href="#">Unbranded Boot</a>	Unbranded Boot can suppress Windows elements that appear when Windows starts or resumes and can suppress the crash screen when Windows encounters an error that it cannot recover from.
<a href="#">Unified Write Filter (UWF)</a>	Use Unified Write Filter (UWF) on your device to help protect your physical storage media, including most standard writable storage types that are supported by Windows, such as physical hard disks, solid-state drives, internal USB devices, external SATA devices, and so on. You can also use UWF to make read-only media appear to the OS as a writable volume.

**TIP**

In addition to the customizations above for OEMs, Windows 10 provides a [Mobile device management \(MDM\)](#) to help IT pros manage company security policies and business applications, while avoiding compromise of the users' privacy on their personal devices. Under MDM, mobile device OEMs can also create custom configuration service providers (CSPs) to manage their devices. For more information, see [Mobile device management](#).

## Related topics

**Keyboard Filter reference:** [Keyboard Filter key names](#)

[Keyboard Filter WMI provider reference](#)

**Shell Launcher reference:** [WESL\\_UserSetting](#)

**Unified Write Filter reference:** [Unified Write Filter WMI provider reference](#)

# WEDL\_AssignedAccess

10/2/2018 • 2 minutes to read • [Edit Online](#)

This Windows Management Instrumentation (WMI) provider class configures settings for assigned access.

## Syntax

```
class WEDL_AssignedAccess {
    [Key] string UserID;
    [Read, Write] string AppUserModelId;
    [Read] sint32 Status;
};
```

## Members

The following tables list any methods and properties that belong to this class.

### Methods

This class contains no methods.

### Properties

PROPERTY	DATA TYPE	QUALIFIERS	DESCRIPTION		
<b>UserID</b>	string	[key]	The security identifier (SID) for the user account that you want to use as the assigned access account.		
<b>AppUserModelId</b>	string	[read, write]	The Application User Model ID (AUMID) of the Windows app to launch for the assigned access account.		
<b>Status</b>	Boolean	none	Indicates the current status of the assigned access configuration: <table><thead><tr><th>VALUE</th><th>DESCRIPTION</th></tr></thead></table>	VALUE	DESCRIPTION
VALUE	DESCRIPTION				

PROPERTY	DATA TYPE	QUALIFIERS	DESCRIPTION VALUE	DESCRIPTION
			0	A valid account is configured, but no Windows app is specified. Assigned access is not enabled.
			1	Assigned access is enabled.
			0x100	UserSID error: cannot find the account.
			0x103	UserSID error: the account profile does not exist.
			0x200	AppUserModelID error: cannot find the Windows app.

PROPERTY	DATA TYPE	QUALIFIERS	DESCRIPTION VALUE	DESCRIPTION
			0x201	Task Scheduler error: Could not schedule task. Make sure that the Task Scheduler service is running.
	0xffffffff	Unspecified error.		

## Remarks

Changes to assigned access do not affect any sessions that are currently signed in; you must sign out and sign back in.

## Example

The following Windows PowerShell script demonstrates how to use this class to set up an assigned access account.

```

#
#---Define variables---
#
$COMPUTER = "localhost"
$NAMESPACE = "root\standardcimv2\embedded"

# Define the assigned access account.
# To use a different account, change $AssignedAccessAccount to a user account that is present on your device.

$AssignedAccessAccount = "KioskAccount"

# Define the Windows app to launch, in this example, use the Application Model User ID (AUMID) for Windows Calculator.
# To use a different Windows app, change $AppAUMID to the AUMID of the Windows app to launch.
# The Windows app must be installed for the account.

$appAUMID = "Microsoft.WindowsCalculator_8wekyb3d8bbwe!App"

#
#---Define helper functions---
#
function Get-UsernameSID($AccountName) {

# This function retrieves the SID for a user account on a machine.
# This function does not check to verify that the user account actually exists.

```

```

$NTUserObject = New-Object System.Security.Principal.NTAccount($AccountName)
$NTUserSID = $NTUserObject.Translate([System.Security.Principal.SecurityIdentifier])

return $NTUserSID.Value
}

#
#---Set up the new assigned access account---
#

# Get the SID for the assigned access account.

$AssignedAccessUserID = Get-UsernameSID($AssignedAccessAccount)

# Check to see if an assigned access account is already set up, and if so, clear it.

$AssignedAccessConfig = get-WMIObject -namespace $NAMESPACE -computer $COMPUTER -class WEDL_AssignedAccess

if ($AssignedAccessConfig) {

# Configuration already exists. Delete it so that we can create a new one, since only one assigned access
account can be set up at a time.

$AssignedAccessConfig.delete()

}

# Configure assigned access to launch the specified Windows app for the specified account.

Set-WmiInstance -class WEDL_AssignedAccess -ComputerName $COMPUTER -Namespace $NAMESPACE -Arguments @{
    UserID = $AssignedAccessUserID;
    AppUserModelId = $AppAUMID
} | Out-Null;

# Confirm that the settings were created properly.

$AssignedAccessConfig = get-WMIObject -namespace $NAMESPACE -computer $COMPUTER -class WEDL_AssignedAccess

if ($AssignedAccessConfig) {

    "Set up assigned access for the " + $AssignedAccessAccount + " account."
    "    UserID = " + $AssignedAccessConfig.UserID
    "    AppModelId = " + $AssignedAccessConfig.AppUserModelId

} else {

    "Could not set up assigned access account."
}

```

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	Yes
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

# Custom Logon

10/2/2018 • 2 minutes to read • [Edit Online](#)

You can use the Custom Logon feature to suppress Windows 10 UI elements that relate to the Welcome screen and shutdown screen. For example, you can suppress all elements of the Welcome screen UI and provide a custom logon UI. You can also suppress the Blocked Shutdown Resolver (BSDR) screen and automatically end applications while the OS waits for applications to close before a shutdown.

Custom Logon settings do not modify the credential behavior of **Winlogon**, so you can use any credential provider that is compatible with Windows 10 to provide a custom sign-in experience for your device.

## Requirements

Windows 10 Enterprise or Windows 10 Education.

## Terminology

**Turn on, enable:** To make the setting available to the device and optionally apply the settings to the device. Generally *turn on* is used in the user interface or control panel, whereas *enable* is used for command line.

**Configure:** To customize the setting or sub-settings.

**Embedded Logon:** This feature is called Embedded Logon in Windows 10, version 1511.

**Custom Logon:** This feature is called Custom Logon in Windows 10, version 1607 and later.

## Turn on Custom Logon

Custom Logon is an optional component and is not turned on by default in Windows 10. It must be turned on prior to configuring. You can turn on and configure Custom Logon in a customized Windows 10 image (.wim) if Microsoft Windows has not been installed. If Windows has already been installed and you are applying a provisioning package to configure Custom Logon, you must first turn on Custom Logon in order for a provisioning package to be successfully applied.

The Custom Logon feature is available in the Control Panel. You can set Custom Logon by following these steps:

### Turn on Custom Logon in Control Panel

1. In the **Search the web and Windows** field, type **Turn Windows features on or off**.
2. In the **Windows Features** window, expand the **Device Lockdown** node, and select or clear the checkbox for **Custom Logon**.

### Turn on and configure Custom Logon using DISM

1. Open a command prompt with administrator rights.
2. Copy install.wim to a temporary folder on hard drive (in the following steps, we'll assume it's called C:\wim).
3. Create a new directory.
4. Mount the image.

```
md c:\wim
```

```
dism /mount-wim /wimfile:c:\bootmedia\sources\install.wim /index:1 /MountDir:c:\wim
```

5. Enable the feature.

```
dism /image:c:\wim /enable-feature /featureName:Client-EmbeddedLogon
```

6. Commit the change.

```
dism /unmount-wim /MountDir:c:\wim /Commit
```

### Configure Custom Logon settings using Unattend

You can configure the Unattend settings in the [Microsoft-Windows-Embedded-EmbeddedLogon](#) component to add custom logon features to your image during the design or imaging phase. You can manually create an Unattend answer file or use Windows System Image Manager (Windows SIM) to add the appropriate settings to your answer file. For more information about the custom logon settings and XML examples, see the settings in Microsoft-Windows-Embedded-EmbeddedLogon.

The following example shows how to disable all Welcome screen UI elements and the **Switch user** button.

```
<settings pass="specialize">
    <component name="Microsoft-Windows-Embedded-EmbeddedLogon" processorArchitecture="x86"
publicKeyToken="31bf3856ad364e35" language="neutral" versionScope="nonSxS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <BrandingNeutral>1</BrandingNeutral>
        <AnimationDisabled>1</AnimationDisabled>
        <NoLockScreen>1</NoLockScreen>
        <UIVerbosityLevel>1</UIVerbosityLevel>
        <HideAutoLogonUI>1</HideAutoLogonUI>
    </component>
</settings>
```

## In this section

- [Complementary features to Custom Logon](#)
- [Troubleshooting Custom Logon](#)

## Related topics

[Unbranded Boot](#)

[Shell Launcher](#)

# Complementary features to Custom Logon

10/2/2018 • 2 minutes to read • [Edit Online](#)

You may want to use or change some of the following features in conjunction with Custom Logon to complete the user experience.

## Power button

We recommend that you remove the power button from the Welcome screen and block the physical power button so that a user cannot turn off the device when using assigned access or Shell Launcher.

Go to **Power Options > Choose what the power button does**, change the setting to **Do nothing**, and then **Save changes**.

## Welcome screen

### To remove buttons from the Welcome screen

- To remove buttons from the Welcome screen, set the appropriate value for **BrandingNeutral** in the following registry key:

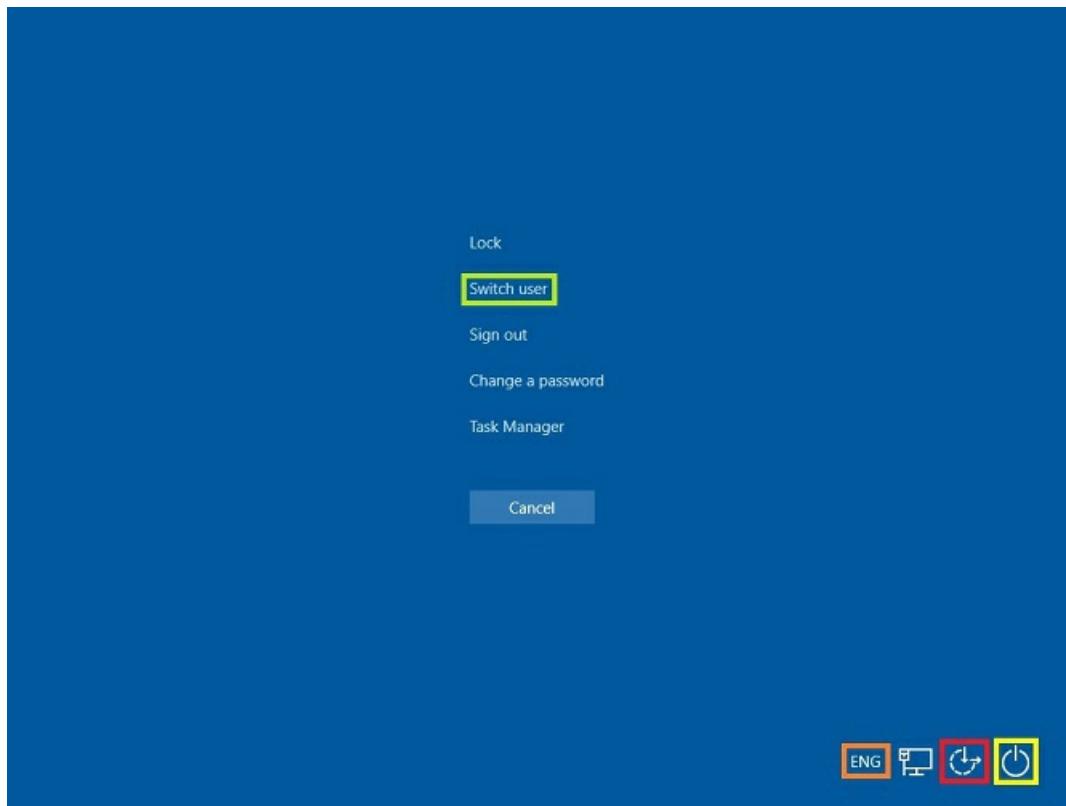
**HKLM\Software\Microsoft\Windows Embedded\EmbeddedLogon**

The following table shows the possible values. To disable multiple Welcome screen UI elements, combine these values using bitwise exclusive-or logic.

ACTION	REGISTRY VALUE
Disable all Welcome screen UI elements	<b>static const DWORD EMBEDDED_DISABLE_LOGON_ANCHOR_ALL = 0x1</b>
Disable the Power button	<b>static const DWORD EMBEDDED_DISABLE_LOGON_ANCHOR_SHUTDOWN = 0x2</b>
Disable the Language button	<b>static const DWORD EMBEDDED_DISABLE_LOGON_ANCHOR_LANGUAGE = 0x4</b>
Disable the Ease of Access button	<b>static const DWORD EMBEDDED_DISABLE_LOGON_ANCHOR_EASEOFACCE SS = 0x8</b>
Disable the Switch user button.	<b>static const DWORD EMBEDDED_DISABLE_BACK_BUTTON = 0x10</b>

ACTION	REGISTRY VALUE
Disable the Blocked Shutdown Resolver (BSDR) screen so that restarting or shutting down the system causes the OS to immediately force close any open applications that are blocking system shut down. No UI is displayed, and users are not given a chance to cancel the shutdown process	<pre>static const DWORD EMBEDDED_DISABLE_BSDR=0x20</pre>

In the following image of the `[ctrl + alt + del]` screen, you can see the Switch user button highlighted by a light green outline, the Language button highlighted by an orange outline, the Ease of Access button highlighted by a red outline, and the power button highlighted by a yellow outline. If you disable these buttons, they are hidden from the UI.



You can remove the Wireless UI option from the Welcome screen by using Group Policy.

#### Remove Wireless UI from the Welcome screen

##### To remove Wireless UI from the Welcome screen

1. From a command prompt, run gpedit.msc to open the Local Group Policy Editor.
2. In the Local Group Policy Editor, under **Computer Configuration**, expand **Administrative Templates**, expand **System**, and then tap or click **Logon**.
3. Double-tap or click **Do not display network selection UI**.

## Related topics

[Custom Logon](#)

[Troubleshooting Custom Logon](#)

# Troubleshooting Custom Logon

10/2/2018 • 2 minutes to read • [Edit Online](#)

This section highlights some common issues that you may encounter when using Custom Logon.

## When automatic sign-in is enabled, the device asks for a password when resuming from sleep or hibernate

This can occur when your device is configured to require a password when waking up from a sleep state.

### To disable password protection on wake up

1. If you have write filters enabled on your device, perform the following steps to disable them so that you can save setting changes:

- At an administrator command prompt, type the following command:

```
ufmgr.exe filter disable
```

- To restart the device, type the following command:

```
ufmgr.exe restart
```

2. In **Control Panel**, search for **Power Options**, and then click the Power Options heading.

3. Under the **Power Options** heading, click **Require a password on wakeup**.

4. On the **Define power buttons and turn on password protection** page, under **Password protection on wakeup**, select **Don't require a password**.

5. If you disabled write filters, perform the following steps to enable them again:

- At an administrator command prompt, type the following command:

```
ufmgr.exe filter enable
```

- To restart the device, type the following command:

```
ufmgr.exe restart
```

## The device displays a black screen during setup

Set the **HideAutoLogonUI** and **AnimationDisabled** settings to **0** (zero). The device will then display a default screen during setup.

## The device displays a black screen when Ctrl+Alt+Del is pressed

**HideAutoLogonUI** and **ForceAutoLogon** have known issues when used together. To avoid a black screen, we recommend you use Keyboard Filter to block this key combination.

## The device displays a black screen when Windows key + L is used to lock the device

**HideAutoLogonUI** and **ForceAutoLogon** have known issues when used together. To avoid a black screen, we recommend you use Keyboard Filter to block this key combination.

### The device displays a black screen when Notepad is opened, any characters are typed and the current user signs out, or the device is rebooted, or the device is shut down

**HideAutoLogonUI** and **ForceAutoLogon** have known issues when used together. To avoid a black screen, we recommend you disable the Blocked Shutdown Resolver Screen (BSDR).

#### WARNING

When the BSDR screen is disabled, restarting or shutting down the device causes the OS to immediately force close any open applications that are blocking system shutdown. No UI is displayed, and users are not given a chance to cancel the shutdown process. This can result in lost data if any open applications have unsaved data.

## The device displays a black screen when the device is suspended and then resumed

**HideAutoLogonUI** and **ForceAutoLogon** have known issues when used together. To avoid a black screen, we recommend you disable the password protection on wakeup.

### To disable password protection on wakeup

1. In **Control Panel**, click **Power Options**.
2. In the **Power Options** item, click **Require a password on wakeup**.
3. On the **Define power buttons and turn on password protection** page, under **Password protection on wakeup**, select **Don't require a password**.

### The device displays a black screen when a password expiration screen is displayed

**HideAutoLogonUI** has a known issue. To avoid a black screen, we recommend you set the password to never expire.

### To set a password to never expire on an individual user account

1. On your device, open a command prompt with administrator privileges.
2. Type the following, replacing <accountname> with the name of the account you want to remove the password expiration from.

```
net accounts <accountname> /expires:never
```

### To set passwords to never expire on all user accounts

1. On your device, open a command prompt with administrator privileges.
2. Type the following

```
net accounts /MaxPWAge:unlimited
```

## Related topics

[Custom Logon](#)

## Complementary features to Custom Logon

# Keyboard Filter

10/2/2018 • 8 minutes to read • [Edit Online](#)

You can use Keyboard Filter to suppress undesirable key presses or key combinations. Normally, a customer can use certain Microsoft Windows key combinations like Ctrl+Alt+Delete or Ctrl+Shift+Tab to alter the operation of a device by locking the screen or using Task Manager to close a running application. This may not be desirable if your device is intended for a dedicated purpose.

The Keyboard Filter feature works with physical keyboards, the Windows on-screen keyboard, and the touch keyboard. Keyboard Filter also detects dynamic layout changes, such as switching from one language set to another, and continues to suppress keys correctly, even if the location of suppressed keys has changed on the keyboard layout.

## NOTE

Keyboard filter is not supported in a remote desktop session.

## Requirements

Windows 10 Enterprise or Windows 10 Education.

## Terminology

- **Turn on, enable:** To make the setting available to the device and optionally apply the settings to the device. Generally *turn on* is used in the user interface or control panel, whereas *enable* is used for command line.
- **Configure:** To customize the setting or sub-settings.
- **Embedded Keyboard Filter:** This feature is called Embedded Keyboard Filter in Windows 10, version 1511.
- **Keyboard Filter:** This feature is called Keyboard Filter in Windows 10, version 1607 and later.

## Turn on Keyboard Filter

By default, Keyboard Filter is not turned on. You can turn Keyboard Filter on or off for your device by using the following steps.

Turning on an off Keyboard Filter requires that you restart your device. Keyboard Filter is automatically enabled after the restart.

### Turn on Keyboard Filter by using Control Panel

1. In the **Search the web and Windows** field, type **Programs and Features** and either press **Enter** or tap or click **Programs and Features** to open it.
2. In the **Programs and Features** window, click **Turn Windows features on or off**.
3. In the **Windows Features** window, expand the **Device Lockdown** node, and select or clear the checkbox for **Keyboard Filter**.
4. Click **OK**. The **Windows Features** window indicates Windows 10 is searching for required files and displays a progress bar. Once found, the window indicates Windows 10 is applying the changes. When completed, the window indicates the requested changes are completed.

5. Click **Close** to close the **Windows Features** window.

### Configure Keyboard using Unattend

1. You can configure the Unattend settings in the [Microsoft-Windows-Embedded-KeyboardFilterService](#) component to add Keyboard Filter features to your image during the design or imaging phase.
2. You can manually create an Unattend answer file or use Windows System Image Manager (Windows SIM) to add the appropriate settings to your answer file. For more information about the keyboard filter settings and XML examples, see the settings in [Microsoft-Windows-Embedded-KeyboardFilterService](#).

### Turn on and configure Keyboard Filter using Windows Configuration Designer

The Keyboard Filter settings are also available as Windows provisioning settings so you can configure these settings to be applied during the image deployment time or runtime. You can set one or all keyboard filter settings by creating a provisioning package using Windows Configuration Designer and then applying the provisioning package during image deployment time or runtime.

1. Build a provisioning package in Windows Configuration Designer by following the instructions in [Create a provisioning package](#).

#### NOTE

In the **Select Windows Edition** window, choose **Common to all Windows desktop editions**.

2. On the **Available customizations** page, select **Runtime settings > SMISettings**, and then set the desired values for the keyboard filter settings.
3. Once you have finished configuring the settings and building the provisioning package, you can apply the package to the image deployment time or runtime. See [Apply a provisioning package](#) for more information. Note that the process for applying the provisioning package to a Windows 10 Enterprise image is the same.

This example uses a Windows image called install.wim, but you can use the same procedure to apply a provisioning package. For more information on DISM, see [What Is Deployment Image Servicing and Management](#).

### Turn on and configure Keyboard Filter by using DISM

1. Open a command prompt with administrator privileges.
2. Copy install.wim to a temporary folder on hard drive (in the following steps, we'll assume it's called C:\wim).
3. Create a new directory.

```
md c:\wim
```

4. Mount the image.

```
dism /mount-wim /wimfile:c:\bootmedia\sources\install.wim /index:1 /MountDir:c:\wim
```

5. Enable the feature.

```
Dism /online /Enable-Feature /FeatureName:Client-KeyboardFilter
```

6. Commit the change.

```
dism /unmount-wim /MountDir:c:\wim /Commit
```

## Keyboard Filter features

Keyboard Filter has the following features:

- Supports hardware keyboards, the standard Windows on-screen keyboard, and the touch keyboard (TabTip.exe).
- Suppresses key combinations even when they come from multiple keyboards.

For example, if a user presses the Ctrl key and the Alt key on a hardware keyboard, while at the same time pressing Delete on a software keyboard, Keyboard Filter can still detect and suppress the Ctrl+Alt+Delete functionality.

- Supports numeric keypads and keys designed to access media player and browser functionality.
- Can configure a key to breakout of a locked down user session to return to the Welcome screen.
- Automatically handles dynamic layout changes.
- Can be enabled or disabled for administrator accounts.
- Can force disabling of Ease of Access functionality.
- Can block physical hardware keys.
- Supports x86 and x64 architectures.

## Keyboard scan codes and layouts

When a key is pressed on a physical keyboard, the keyboard sends a scan code to the keyboard driver. The driver then sends the scan code to the OS and the OS converts the scan code into a virtual key based on the current active layout. The layout defines the mapping of keys on the physical keyboard, and has many variants. A key on a keyboard always sends the same scan code when pressed, however this scan code can map to different virtual keys for different layouts. For example, in the English (United States) keyboard layout, the key to the right of the P key maps to "{". However, in the Swedish (Sweden) keyboard layout, the same key maps to "Å".

Keyboard Filter can block keys either by the scan code or the virtual key. Blocking keys by the scan code is useful for custom keyboards that have special scan codes that do not translate into any single virtual key. Blocking keys by the virtual key is generally more convenient because it is easier to read and Keyboard Filter suppresses the key correctly even when the location of the key changes because of a layout change.

When you configure Keyboard Filter to block keys by using the virtual key, you must use the English names for the virtual keys. For more information about the names of the virtual keys, see [keyboard filter key names](#).

For the Windows on-screen keyboard, keyboard filter converts each keystroke into a scan code based on the layout, and back into a virtual key. This allows keyboard filter to suppress the on-screen keyboard keys in the same manner as physical keyboard keys, whether they are configured by scan code or virtual key.

## Keyboard Filter and ease of access features

By default, ease of access features are enabled and Keyboard Filter is disabled for administrator accounts.

If Sticky Keys are enabled, a user can bypass Keyboard Filter in certain situations. You can configure keyboard filter to disable all ease of access features and prevent users from enabling them.

You can enable ease of access features for administrator accounts, while still disabling them for standard user accounts, by making sure that Keyboard Filter is disabled for administrator accounts.

## Keyboard Filter configuration

You can configure the following options for Keyboard Filter:

- Set/unset predefined key combinations to be suppressed.
- Add/remove custom defined key combinations to be suppressed.
- Enable/disable keyboard filter for administrator accounts.
- Force disabling ease of access features.
- Configure a breakout key sequence to break out of a locked down account.

Most configuration changes take effect immediately. Some changes, such as enabling or disabling Keyboard Filter for administrators, do not take effect until the user signs out of the account and then back in. If you change the breakout key scan code, you must restart the device before the change takes effect.

You can configure keyboard filter by using Windows Management Instrumentation (WMI) providers. You can use the Keyboard Filter WMI providers directly in a PowerShell script or in an application.

For more information about Keyboard Filter WMI providers, see [Keyboard Filter WMI provider reference](#).

## Keyboard breakout

You may need to sign in to a locked down device with a different account in order to service or configure the device. You can configure a breakout key to break out of a locked down account by specifying a key scan code. When you press Ctrl+Alt+Delete, Windows presents the Welcome screen so that you can sign in to a different account.

The breakout key is set to the scan code for the left Windows logo key by default. You can use the [WEKF\\_Settings](#) WMI class to change the breakout key scan code. If you change the breakout key scan code, you must restart the device before the change takes effect.

## Keyboard Filter considerations

Starting a device in Safe Mode bypasses keyboard filter. The Keyboard Filter service is not loaded in Safe Mode, and keys are not blocked in Safe Mode.

Keyboard filter cannot block the Sleep key.

Some hardware keys, such as rotation lock, do not have a defined virtual key. You can still block these keys by using the scan code of the key.

The add (+), multiply (\*), subtract (-), divide (/), and decimal (.) keys have different virtual keys and scan codes on the numeric keypad than on the main keyboard. You must block both keys to block these keys. For example, to block the multiply key, you must add a rule to block "\*" as well as a rule to block Multiply.

When locking the screen by using the on-screen keyboard, or a combination of a physical keyboard and the on-screen keyboard, the on-screen keyboard sends an additional Windows logo key keystroke to the OS. If your device is using the Windows 10 shell and you use keyboard filter to block Windows logo key+L, the extra Windows logo key keystroke causes the shell to switch between the **Start** screen and the last active app when a user attempts to lock the device by using the on-screen keyboard, which may be unexpected behavior.

Some custom keyboard software, such as Microsoft IntelliType Pro, can install Keyboard Filter drivers that prevent Keyboard Filter from being able to block some or all keys, typically extended keys like BrowserHome and Search.

## In this section

- [Keyboard Filter key names](#)
- [Predefined key combinations](#)
- [Keyboard Filter WMI provider reference](#)
- [Windows PowerShell script samples for Keyboard Filter](#)



# Keyboard Filter key names

10/2/2018 • 4 minutes to read • [Edit Online](#)

You can configure Keyboard Filter to block keys or key combinations. A key combination consists of one or more modifier keys, separated by a plus sign (+), and either a key name or a key scan code. In addition to the keys listed in the tables below, you can also use the predefined key combinations names as custom key combinations, but we recommend using the predefined key settings when enabling or disabling predefined key combinations.

The key names are grouped as follows:

- [Modifier keys](#)
- [System keys](#)
- [Cursor and math keys](#)
- [State keys](#)
- [OEM keys](#)
- [Function keys](#)

## Modifier keys

You can use the modifier keys listed in the following table when you configure keyboard filter. Multiple modifiers must be separated by a plus sign (+). You can also configure Keyboard Filter to block any modifier key even if it's not part of a key combination..

MODIFIER KEY NAME	VIRTUAL KEY	DESCRIPTION
<b>Ctrl</b>	VK_CONTROL	The Ctrl key
<b>LCtrl</b>	VK_LCONTROL	The left Ctrl key
<b>RCtrl</b>	VK_RCONTROL	The right Ctrl key
<b>Control</b>	VK_CONTROL	The Ctrl key
<b>LControl</b>	VK_LCONTROL	The left Ctrl key
<b>RControl</b>	VK_RCONTROL	The right Ctrl key
<b>Alt</b>	VK_MENU	The Alt key
<b>LAlt</b>	VK_LMENU	The left Alt key
<b>RAlt</b>	VK_RMENU	The right Alt key

MODIFIER KEY NAME	VIRTUAL KEY	DESCRIPTION
<b>Shift</b>	VK_SHIFT	The Shift key
<b>LShift</b>	VK_LSHIFT	The left Shift key
<b>RShift</b>	VK_RSHIFT	The right Shift key
<b>Win</b>	VK_WIN	The Windows logo key
<b>LWin</b>	VK_LWIN	The left Windows logo key
<b>RWin</b>	VK_RWIN	The right Windows logo key
<b>Windows</b>	VK_WIN	The Windows logo key
<b>LWindows</b>	VK_LWIN	The left Windows logo key
<b>RWindows</b>	VK_RWIN	The right Windows logo key

## System keys

KEY NAME	VIRTUAL KEY	DESCRIPTION
<b>Backspace</b>	VK_BACK	The Backspace key
<b>Back</b>	VK_BACK	The Backspace key
<b>Tab</b>	VK_TAB	The Tab key
<b>Clear</b>	VK_CLEAR	The Clear key
<b>Enter</b>	VK_RETURN	The Enter key
<b>Return</b>	VK_RETURN	The Enter key
<b>Pause</b>	VK_PAUSE	The Pause key
<b>Esc</b>	VK_ESCAPE	The Esc key

KEY NAME	VIRTUAL KEY	DESCRIPTION
<b>Escape</b>	VK_ESCAPE	The Esc key
<b>Space</b>	VK_SPACE	The Spacebar
<b>Break</b>	VK_BREAK	The Break key
<b>Select</b>	VK_SELECT	The Select key
<b>PrintScreen</b>	VK_PRINT	The Print Screen key
<b>PrintScrn</b>	VK_PRINT	The Print Screen key
<b>Print</b>	VK_PRINT	The Print Screen key
<b>Execute</b>	VK_EXECUTE	The Execute key
<b>Snapshot</b>	VK_SNAPSHOT	The Print Screen key
<b>Help</b>	VK_HELP	The Help key

## Cursor and edit keys

KEY NAME	VIRTUAL KEY	DESCRIPTION
<b>PageUp</b>	VK_PRIOR	The Page Up key
<b>Prior</b>	VK_PRIOR	The Page Up key
<b>PgUp</b>	VK_PRIOR	The Page Up key
<b>PageDown</b>	VK_NEXT	The Page Down key
<b>PgDown</b>	VK_NEXT	The Page Down key
<b>Next</b>	VK_NEXT	The Page Down key
<b>End</b>	VK_END	The End key

KEY NAME	VIRTUAL KEY	DESCRIPTION
<b>Home</b>	VK_HOME	The Home key
<b>Left</b>	VK_LEFT	The Left Arrow key
<b>Up</b>	VK_UP	The Up Arrow key
<b>Right</b>	VK_RIGHT	The Right Arrow key
<b>Down</b>	VK_DOWN	The Down Arrow key
<b>Insert</b>	VK_INSERT	The Insert key
<b>Delete</b>	VK_DELETE	The Delete key
<b>Del</b>	VK_DELETE	The Delete key
<b>Separator</b>	VK_SEPARATOR	The Separator key

## State keys

KEY NAME	VIRTUAL KEY	DESCRIPTION
<b>NumLock</b>	VK_NUMLOCK	The Num Lock key
<b>ScrollLock</b>	VK_SCROLL	The Scroll Lock key
<b>Scroll</b>	VK_SCROLL	The Scroll Lock key
<b>CapsLock</b>	VK_CAPITAL	The Caps Lock key
<b>Capital</b>	VK_CAPITAL	The Caps Lock key

## OEM keys

KEY NAME	VIRTUAL KEY	DESCRIPTION
<b>KeypadEqual</b>	VK_OEM_NECK_EQUAL	The Equal Sign (=) key on the numeric keypad (OEM-specific)

KEY NAME	VIRTUAL KEY	DESCRIPTION
<b>Dictionary</b>	VK_OEM_FJ_JISHO	The Dictionary key (OEM-specific)
<b>Unregister</b>	VK_OEM_FJ_MASSHOU	The Unregister Word key (OEM-specific)
<b>Register</b>	VK_OEM_FJ_TOUROKU	The Register Word key (OEM-specific)
<b>LeftOyayubi</b>	VK_OEM_FJ_LOYA	The Left OYAYUBI key (OEM-specific)
<b>RightOyayubi</b>	VK_OEM_FJ_ROYA	The Right OYAYUBI key (OEM-specific)
<b>OemPlus</b>	VK_OEM_PLUS	For any country/region, the Plus Sign (+') key
<b>OemComma</b>	VK_OEM_COMMA	For any country/region, the Comma (,) key
<b>OemMinus</b>	VK_OEM_MINUS	For any country/region, the Minus Sign (-) key
<b>OemPeriod</b>	VK_OEM_PERIOD	For any country/region, the Period (.) key
<b>Oem1</b>	VK_OEM_1	Varies by keyboard
<b>Oem2</b>	VK_OEM_2	Varies by keyboard
<b>Oem3</b>	VK_OEM_3	Varies by keyboard
<b>Oem4</b>	VK_OEM_4	Varies by keyboard
<b>Oem5</b>	VK_OEM_5	Varies by keyboard
<b>Oem6</b>	VK_OEM_6	Varies by keyboard
<b>Oem7</b>	VK_OEM_7	Varies by keyboard

KEY NAME	VIRTUAL KEY	DESCRIPTION
<b>Oem8</b>	VK_OEM_8	Varies by keyboard
<b>OemAX</b>	VK_OEM_AX	The AX key on a Japanese AX keyboard
<b>Oem102</b>	VK_OEM_102	Either the angle bracket key or the backslash key on the RT 102-key keyboard

## Function keys

KEY NAME	VIRTUAL KEY	DESCRIPTION
<b>F1</b>	VK_F1	The F1 key
<b>F2</b>	VK_F2	The F2 key
<b>F3</b>	VK_F3	The F3 key
<b>F4</b>	VK_F4	The F4 key
<b>F5</b>	VK_F5	The F5 key
<b>F6</b>	VK_F6	The F6 key
<b>F7</b>	VK_F7	The F7 key
<b>F8</b>	VK_F8	The F8 key
<b>F9</b>	VK_F9	The F9 key
<b>F10</b>	VK_F10	The F10 key
<b>F11</b>	VK_F11	The F11 key
<b>F12</b>	VK_F12	The F12 key
<b>F13</b>	VK_F13	The F13 key

KEY NAME	VIRTUAL KEY	DESCRIPTION
F14	VK_F14	The F14 key
F15	VK_F15	The F15 key
F16	VK_F16	The F16 key
F17	VK_F17	The F17 key
F18	VK_F18	The F18 key
F19	VK_F19	The F19 key
F20	VK_F20	The F20 key
F21	VK_F21	The F21 key
F22	VK_F22	The F22 key
F23	VK_F23	The F23 key
F24	VK_F24	The F24 key

## Numeric keypad keys

KEY NAME	VIRTUAL KEY	DESCRIPTION
Numpad0	VK_NUMPAD0	The 0 key on the numeric keypad
Numpad1	VK_NUMPAD1	The 1 key on the numeric keypad
Numpad2	VK_NUMPAD2	The 2 key on the numeric keypad
Numpad3	VK_NUMPAD3	The 3 key on the numeric keypad
Numpad4	VK_NUMPAD4	The 4 key on the numeric keypad
Numpad5	VK_NUMPAD5	The 5 key on the numeric keypad

KEY NAME	VIRTUAL KEY	DESCRIPTION
<b>Numpad6</b>	VK_NUMPAD6	The 6 key on the numeric keypad
<b>Numpad7</b>	VK_NUMPAD7	The 7 key on the numeric keypad
<b>Numpad8</b>	VK_NUMPAD8	The 8 key on the numeric keypad
<b>Numpad9</b>	VK_NUMPAD9	The 9 key on the numeric keypad
<b>Multiply</b>	VK_MULTIPLY	The Multiply (*) key on the numeric keypad
<b>Add</b>	VK_ADD	The Add (+) key on the numeric keypad
<b>Subtract</b>	VK_SUBTRACT	The Subtract (-) key on the numeric keypad
<b>Decimal</b>	VK_DECIMAL	The Decimal (.) key on the numeric keypad
<b>Divide</b>	VK_DIVIDE	The Divide (/) key on the numeric keypad

## Related topics

[Keyboard filter](#)

# Predefined key combinations

10/2/2018 • 4 minutes to read • [Edit Online](#)

This topic lists the key combinations that are predefined by a keyboard filter.

You can use the values in the WEKF\_PredefinedKey.Id column to configure the Windows Management Instrumentation (WMI) class [WEKF\\_PredefinedKey](#).

## Accessibility keys

The following table contains predefined key combinations for accessibility:

KEY COMBINATION	WEKF_PREDEFINEDKEY.ID	BLOCKED BEHAVIOR
Left Alt + Left Shift + Print Screen	<b>LShift+LAlt+PrintScrn</b>	Open High Contrast.
Left Alt + Left Shift + Num Lock	<b>LShift+LAlt+NumLock</b>	Open Mouse Keys.
Windows logo key + U	<b>Win+U</b>	Open Ease of Access Center.

## Application keys

The following table contains predefined key combinations for controlling application state:

KEY COMBINATION	WEKF_PREDEFINEDKEY.ID	BLOCKED BEHAVIOR
Alt + F4	<b>Alt+F4</b>	Close application.
Ctrl + F4	<b>Ctrl+F4</b>	Close window.
Windows logo key + F1	<b>Win+F1</b>	Open Windows Help.

## Shell keys

The following table contains predefined key combinations for general UI control:

KEY COMBINATION	WEKF_PREDEFINEDKEY.ID	BLOCKED BEHAVIOR
Alt + Spacebar	<b>Alt+Space</b>	Open shortcut menu for the active window.
Ctrl + Esc	<b>Ctrl+Esc</b>	Open the Start screen.
Ctrl + Windows logo key + F	<b>Ctrl+Win+F</b>	Open Find Computers.
Windows logo key + Break	<b>Win+Break</b>	Open System dialog box.
Windows logo key + E	<b>Win+E</b>	Open Windows Explorer.

KEY COMBINATION	WEKF_PREDEFINEDKEY.ID	BLOCKED BEHAVIOR
Windows + F	<b>Win+F</b>	Open Search.
Windows logo key + P	<b>Win+P</b>	Cycle through Presentation Mode. Also blocks the Windows logo key + Shift + P and the Windows logo key + Ctrl + P key combinations.
Windows logo key + R	<b>Win+R</b>	Open Run dialog box.
Alt + Tab	<b>Alt+Tab</b>	Switch task. Also blocks the Alt + Shift + Tab key combination.
Ctrl + Tab	<b>Ctrl+Tab</b>	Switch window.
Windows logo key + Tab	<b>Win+Tab</b>	Cycle through Microsoft Store apps. Also blocks the Windows logo key + Ctrl + Tab and Windows logo key + Shift + Tab key combinations.
Windows logo key + D	<b>Win+D</b>	Show desktop.
Windows logo key + M	<b>Win+M</b>	Minimize all windows.
Windows logo key + Home	<b>Win+Home</b>	Minimize or restore all inactive windows.
Windows logo key + T	<b>Win+T</b>	Set focus on taskbar and cycle through programs.
Windows logo key + B	<b>Win+B</b>	Set focus in the notification area.
Windows logo key + Minus Sign	<b>Win+-</b>	Zoom out.
Windows logo key + Plus Sign	<b>Win++</b>	Zoom in.
Windows logo key + Esc	<b>Win+Esc</b>	Close Magnifier application.
Windows logo key + Up Arrow	<b>Win+Up</b>	Maximize the active window.
Windows logo key + Down Arrow	<b>Win+Down</b>	Minimize the active window.
Windows logo key + Left Arrow	<b>Win+Left</b>	Snap the active window to the left half of screen.
Windows logo key + Right Arrow	<b>Win+Right</b>	Snap the active window to the right half of screen.
Windows logo key + Shift + Up Arrow	<b>Win+Shift+Up</b>	Maximize the active window vertically.
Windows logo key + Shift + Down Arrow	<b>Win+Shift+Down</b>	Minimize the active window.

KEY COMBINATION	WEKF_PREDEFINEDKEY.ID	BLOCKED BEHAVIOR
Windows logo key + Shift + Left Arrow	<b>Win+Shift+Left</b>	Move the active window to left monitor.
Windows logo key + Shift + Right Arrow	<b>Win+Shift+Right</b>	Move the active window to right monitor.
Windows logo key + Spacebar	<b>Win+Space</b>	Switch layout.
Windows logo key + O	<b>Win+O</b>	Lock device orientation.
Windows logo key + Ctrl + Enter	<b>Win+Ctrl+Enter</b>	Start Narrator.
Windows logo key + Page Up	<b>Win+PageUp</b>	Move a Microsoft Store app to the left monitor.
Windows logo key + Page Down	<b>Win+PageDown</b>	Move a Microsoft Store app to right monitor.
Windows logo key + Period	<b>Win+.</b>	Snap the current screen to the left or right gutter. Also blocks the Windows logo key + Shift + Period key combination.
Windows logo key + C	<b>Win+C</b>	Activate Cortana in listening mode (after user has enabled the shortcut through the UI).
Windows logo key + I	<b>Win+I</b>	Open Settings charm.
Windows logo key + K	<b>Win+K</b>	Open Connect charm.
Windows logo key + H	<b>Win+H</b>	Start dictation.
Windows logo key + Q	<b>Win+Q</b>	Open Search charm.
Windows logo key + W	<b>Win+W</b>	Open Windows Ink workspace.
Windows logo key + Z	<b>Win+Z</b>	Open app bar.
Windows logo key + /	<b>Win+/-</b>	Open input method editor (IME).
Windows logo key + J	<b>Win+J</b>	Swap between snapped and filled applications.
Windows logo key + Comma	<b>Win+,</b>	Peek at the desktop.
Windows logo key + V	<b>Win+V</b>	Cycle through toasts in reverse order.

## Modifier keys

The following table contains predefined key combinations for modifier keys (such as Shift and Ctrl):

KEY COMBINATION	WEKF_PREDEFINEDKEY.ID	BLOCKED KEY
Alt	<b>Alt</b>	Both Alt keys
Application	<b>Application</b>	Application key
Ctrl	<b>Ctrl</b>	Both Ctrl keys
Shift	<b>Shift</b>	Both Shift keys
Windows logo key	<b>Windows</b>	Both Windows logo keys

## Security keys

The following table contains predefined key combinations for OS security:

KEY COMBINATION	WEKF_PREDEFINEDKEY.ID	BLOCKED BEHAVIOR
Ctrl + Alt + Delete	<b>Ctrl+Alt+Del</b>	Open the Windows Security screen.
Ctrl + Shift + Esc	<b>Shift+Ctrl+Esc</b>	Open Task Manager.
Windows logo key + L	<b>Win+L</b>	Lock the device.

## Extended shell keys

The following table contains predefined key combinations for extended shell functions (such as automatically opening certain apps):

KEY COMBINATION	WEKF_PREDEFINEDKEY.ID	BLOCKED KEY
LaunchMail	<b>LaunchMail</b>	Start Mail key
LaunchMediaSelect	<b>LaunchMediaSelect</b>	Select Media key
LaunchApp1	<b>LaunchApp1</b>	Start Application 1 key
LaunchApp2	<b>LaunchApp2</b>	Start Application 2 key

## Browser keys

The following table contains predefined key combinations for controlling the browser:

KEY COMBINATION	WEKF_PREDEFINEDKEY.ID	BLOCKED KEY
BrowserBack	<b>BrowserBack</b>	Browser Back key
BrowserForward	<b>BrowserForward</b>	Browser Forward key
BrowserRefresh	<b>BrowserRefresh</b>	Browser Refresh key

KEY COMBINATION	WEKF_PREDEFINEDKEY.ID	BLOCKED KEY
BrowserStop	<b>BrowserStop</b>	Browser Stop key
BrowserSearch	<b>BrowserSearch</b>	Browser Search key
BrowserFavorites	<b>BrowserFavorites</b>	Browser Favorites key
BrowserHome	<b>BrowserHome</b>	Browser Start and Home key

## Media keys

The following table contains predefined key combinations for controlling media playback:

KEY COMBINATION	WEKF_PREDEFINEDKEY.ID	BLOCKED KEY
VolumeMute	<b>VolumeMute</b>	Volume Mute key
VolumeDown	<b>VolumeDown</b>	Volume Down key
VolumeUp	<b>VolumeUp</b>	Volume Up key
MediaNext	<b>MediaNext</b>	Next Track key
MediaPrev	<b>MediaPrev</b>	Previous Track key
MediaStop	<b>MediaStop</b>	Stop Media key
MediaPlayPause	<b>MediaPlayPause</b>	Play/Pause Media key

## Microsoft Surface keyboard keys

The following table contains predefined key combinations for Microsoft Surface devices:

KEY COMBINATION	WEKF_PREDEFINEDKEY.ID	BLOCKED KEY
Left Alt + Windows logo key	<b>AltWin</b>	Share key
Left Ctrl + Windows logo key	<b>CtrlWin</b>	Devices key
Left Shift + Windows logo key	<b>ShiftWin</b>	Search key
F21	<b>F21</b>	Settings key

## Related topics

[Keyboard filter](#)

# Keyboard Filter WMI provider reference

10/2/2018 • 2 minutes to read • [Edit Online](#)

Describes the Windows Management Instrumentation (WMI) provider classes that you use to configure Keyboard Filter during run time.

[WEKF\\_CustomKey](#) Blocks or unblocks custom defined key combinations.

[WEKF\\_PredefinedKey](#) Blocks or unblocks predefined key combinations.

[WEKF\\_Scancode](#) Blocks or unblocks key combinations by using keyboard scan codes.

[WEKF\\_Settings](#) Enables or disables settings for Keyboard Filter.

## Related topics

[Keyboard filter](#)

# WEKF\_CustomKey

10/2/2018 • 2 minutes to read • [Edit Online](#)

Adds or removes custom-defined key combinations.

## Syntax

```
class WEKF_CustomKey {
    [Static] uint32 Add(
        [In] string CustomKey
    );
    [Static] uint32 Remove(
        [In] string CustomKey
    );

    [Key] string Id;
    [Read, Write] boolean Enabled;
};
```

## Members

The following tables list any methods and properties that belong to this class.

### Methods

METHODS	DESCRIPTION
<a href="#">WEKF_CustomKey.Add</a>	Creates a new custom key combination and enables Keyboard Filter to block the new key combination.
<a href="#">WEKF_CustomKey.Remove</a>	Removes the specified custom key combination. Keyboard Filter stops blocking the key combination that was removed.

### Properties

PROPERTY	DATA TYPE	QUALIFIERS	DESCRIPTION
<b>Id</b>	string	[key]	The name of the custom key combination.

PROPERTY	DATA TYPE	QUALIFIERS	DESCRIPTION						
<b>Enabled</b>	Boolean	[read, write]	<p>Indicates if the key is blocked or unblocked. This property can be one of the following values:</p> <table border="1"> <thead> <tr> <th>VALUE</th><th>DESCRIPTION</th></tr> </thead> <tbody> <tr> <td><b>true</b></td><td>Indicates that the key is blocked.</td></tr> <tr> <td><b>false</b></td><td>Indicates that the key is not blocked.</td></tr> </tbody> </table>	VALUE	DESCRIPTION	<b>true</b>	Indicates that the key is blocked.	<b>false</b>	Indicates that the key is not blocked.
VALUE	DESCRIPTION								
<b>true</b>	Indicates that the key is blocked.								
<b>false</b>	Indicates that the key is not blocked.								

## Remarks

You can specify key combinations by including the modifier keys in the name. The most common modifier names are "Ctrl", "Shift", "Alt", and "Win". You cannot block a combination of non-modifier keys. For example, you can block a key combination of "Ctrl+Shift+F", but you cannot block a key combination of "A+D".

When you block a shift-modified key, you must enter the key as "Shift" + the unmodified key. For example, to block the % key on an English keyboard layout, you must specify the key as "Shift+5". Attempting to block "%", results in Keyboard Filter blocking "5" instead.

When you specify the key combination to block, you must use the English names for the keys. For a list of the key names you can specify, see [Keyboard Filter key names](#).

## Example

The following code demonstrates how to add or enable a custom key combination that Keyboard Filter will block by using the Windows Management Instrumentation (WMI) providers for Keyboard Filter. This example modifies the properties directly and does not call any of the methods defined in [WEKF\\_CustomKey](#).

```

<#
.Synopsis
    This script shows how to use the WMI provider to enable and add
    Keyboard Filter rules through Windows PowerShell on the local computer.
.Parameter ComputerName
    Optional parameter to specify a remote machine that this script should
    manage. If not specified, the script will execute all WMI operations
    locally.
#>
param (
    [String] $ComputerName
)

$CommonParams = @{"namespace"="root\standardcimv2\embedded"}
$CommonParams += $PSBoundParameters

function Enable-Custom-Key($Id) {
    <#
    .Synopsis
        Toggle on a Custom Key Keyboard Filter Rule
    .Description
        Use Get-WMIOBJECT to enumerate all WEKF_CustomKey instances,
        filter against key value "Id", and set that instance's "Enabled"
        property to 1/true.

        In the case that the Custom instance does not exist, add a new
        instance of WEKF_CustomKey using Set-WMIInstance.
    .Example
        Enable-Custom-Key "Ctrl+V"

        Enable filtering of the Ctrl + V sequence.
#>

    $custom = Get-WMIOBJECT -class WEKF_CustomKey @CommonParams |
        where {
            $_.Id -eq "$Id"
        };

    if ($custom) {
        # Rule exists. Just enable it.
        $custom.Enabled = 1;
        $custom.Put() | Out-Null;
        "Enabled Custom Filter $Id.";

    } else {
        Set-WMIInstance `-
            -class WEKF_CustomKey `-
            -argument @{Id="$Id"} `-
            @CommonParams | Out-Null

        "Added Custom Filter $Id.";
    }
}

# Some example uses of the function defined above.

Enable-Custom-Key "Ctrl+V"
Enable-Custom-Key "NumPad0"
Enable-Custom-Key "Shift+NumPad1"

```

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[Keyboard Filter WMI provider reference](#)

[Keyboard Filter key names](#)

# WEKF\_CustomKey.Add

10/2/2018 • 2 minutes to read • [Edit Online](#)

Creates a new custom key combination and enables Keyboard Filter to block the new key combination.

## Syntax

```
[Static] uint32 Add(  
    [In] string CustomKey  
)
```

## Parameters

### *CustomKey*

[in] The custom key combination to add. For a list of valid key names, see [Keyboard Filter key names](#).

## Return Value

Returns an HRESULT value that indicates a [WMI Non-Error Constant](#) or a [WMI Error Constant](#).

## Remarks

**WEKF\_CustomKey.Add** creates a new **WEKF\_CustomKey** object and sets the **Enabled** property of the new object to **true**, and the **Id** property to *CustomKey*.

If a **WEKF\_CustomKey** object already exists with the **Id** property equal to *CustomKey*, then **WEKF\_CustomKey.Add** returns an error code and does not create a new object or modify any properties of the existing object. If the existing **WEKF\_CustomKey** object has the **Enabled** property set to **false**, Keyboard Filter does not block the custom key combination.

## Example

The following code demonstrates how to add or enable a custom key that Keyboard Filter will block by using the Windows Management Instrumentation (WMI) providers for Keyboard Filter.

```

$COMPUTER = "localhost"
$NAMESPACE = "root\standardcimv2\embedded"

# Create a handle to the class instance so we can call the static methods
$classCustomKey = [wmiclass]"\\$COMPUTER\$NAMESPACE":WEKF_CustomKey"

# Create a function to add or enable a key combination for Keyboard Filter to block
function Enable-Custom-Key($KeyId) {

    # Check to see if the custom key object already exists
    $objCustomKey = Get-WMIObject -namespace $NAMESPACE -class WEKF_CustomKey |
        where {$_.Id -eq "$KeyId"};

    if ($objCustomKey) {

        # The custom key already exists, so just enable it
        $objCustomKey.Enabled = 1;
        $objCustomKey.Put() | Out-Null;
        "Enabled ${KeyId}.";

    } else {

        # Create a new custom key object by calling the static Add method
        $retval = $classCustomKey.Add($KeyId);

        # Check the return value to verify that the Add is successful
        if ($retval.ReturnValue -eq 0) {
            "Added ${KeyID}."
        } else {
            "Unknown Error: " + "{0:x0}" -f $retval.ReturnValue
        }
    }
}

# Enable Keyboard Filter to block several custom keys

Enable-Custom-Key "Ctrl+v"
Enable-Custom-Key "Ctrl+v"
Enable-Custom-Key "Shift+4"
Enable-Custom-Key "Ctrl+Alt+w"

# List all the currently existing custom keys

$objCustomKeyList = get-WMIObject -namespace $NAMESPACE -class WEKF_CustomKey
foreach ($objCustomKeyItem in $objCustomKeyList) {
    "Custom key: " + $objCustomKeyItem.Id
    "    enabled: " + $objCustomKeyItem.Enabled
}

```

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[WEKF\\_CustomKey](#)

[Keyboard Filter](#)

# WEKF\_CustomKey.Remove

10/2/2018 • 2 minutes to read • [Edit Online](#)

Removes a custom key combination, causing Keyboard Filter to stop blocking the removed key combination.

## Syntax

```
[Static] uint32 Remove(  
    [In] string CustomKey  
)
```

## Parameters

*CustomKey* [in] The custom key combination to remove.

## Return Value

Returns an HRESULT value that indicates [WMI status](#) or a [WMI error](#).

## Remarks

**WEKF\_CustomKey.Remove** removes an existing **WEKF\_CustomKey** object. If the object does not exist, **WEKF\_CustomKey.Remove** returns an error with the value 0x8007007B.

Because this method is static, you cannot call it on an object instance, but must instead call it at the class level.

## Example

The following code demonstrates how to remove a custom key from Keyboard Filter so it is no longer blocked by using the Windows Management Instrumentation (WMI) providers for Keyboard Filter.

```

$COMPUTER = "localhost"
$NAMESPACE = "root\standardcimv2\embedded"

# Create a handle to the class instance so we can call the static methods
$classCustomKey = [wmiclass]"\\$COMPUTER\$NAMESPACE":WEKF_CustomKey

# Create a function to remove a key combination
function Remove-Custom-Key($KeyId) {

    # Call the static Remove() method on the class reference
    $retval = $classCustomKey.Remove($KeyId)

    # Check the return value for status
    if ($retval.ReturnValue -eq 0) {

        # Custom key combination removed successfully
        "Removed ${KeyId}."

    } elseif ($retval.ReturnValue -eq 2147942523) {

        # No object exists with the specified custom key
        "Failed to remove ${KeyId}. No object found."

    } else {

        # Unknown error, report error code in hexadecimal
        "Failed to remove ${KeyId}. Unknown Error: " + "{0:x0}" -f $retval.ReturnValue
    }
}

# Example of removing a custom key so that Keyboard Filter stops blocking it
Remove-Custom-Key "Ctrl+Alt+w"

# Example of removing all custom keys that have the Enabled property set to false
$objDisabledCustomKeys = Get-WmiObject -Namespace $NAMESPACE -Class WEKF_CustomKey;

foreach ($objCustomKey in $objDisabledCustomKeys) {
    if (!$objCustomKey.Enabled) {
        Remove-Custom-Key($objCustomKey.Id);
    }
}

```

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[WEKF\\_CustomKey](#)

[Keyboard Filter](#)

# WEKF\_PredefinedKey

10/2/2018 • 2 minutes to read • [Edit Online](#)

This class blocks or unblocks predefined key combinations, such as Ctrl+Alt+Delete.

## Syntax

```
class WEKF_PredefinedKey {
    [Static] uint32 Enable (
        [In] string PredefinedKey
    );
    [Static] uint32 Disable (
        [In] string PredefinedKey
    );

    [Key] string Id;
    [Read, Write] boolean Enabled;
};
```

## Members

The following tables list any constructors, methods, fields, and properties that belong to this class.

### Methods

METHODS	DESCRIPTION
<a href="#">WEKF_PredefinedKey.Enable</a>	Blocks the specified predefined key.
<a href="#">WEKF_PredefinedKey.Disable</a>	Unblocks the specified predefined key.

### Properties

PROPERTY	DATA TYPE	QUALIFIERS	DESCRIPTION
<b>Id</b>	string	[key]	The name of the predefined key combination.
<b>Enabled</b>	Boolean	[read, write]	Indicates whether the key is blocked or unblocked. To indicate that the key is blocked, specify <b>true</b> . To indicate that the key is not blocked, specify <b>false</b> .

### Remarks

All accounts have read access to the **WEKF\_PRedefinedKey** class, but only administrator accounts can modify the class.

For a list of predefined key combinations for Keyboard Filter, see [Predefined key combinations](#).

## Example

The following sample Windows PowerShell script blocks the Ctrl+Alt+Delete and the Ctrl+Esc key combinations when the Keyboard Filter service is running.

```
<#
.Synopsis
    This script shows how to use the built in WMI providers to enable and add
    Keyboard Filter rules through Windows PowerShell on the local computer.
.Parameter ComputerName
    Optional parameter to specify a remote machine that this script should
    manage. If not specified, the script will execute all WMI operations
    locally.
#>
param (
    [String] $ComputerName
)

$CommonParams = @{"namespace"="root\standardcimv2\embedded"}
$CommonParams += $PSBoundParameters

function Enable-Predefined-Key($Id) {
    <#
    .Synopsis
        Toggle on a Predefined Key Keyboard Filter Rule
    .Description
        Use Get-WMIOBJECT to enumerate all WEKF_PredefinedKey instances,
        filter against key value "Id", and set that instance's "Enabled"
        property to 1/true.
    .Example
        Enable-Predefined-Key "Ctrl+Alt+Delete"

        Enable CAD filtering
#>

    $predefined = Get-WMIOBJECT -class WEKF_PredefinedKey @CommonParams |
        where {
            $_.Id -eq "$Id"
        };

    if ($predefined) {
        $predefined.Enabled = 1;
        $predefined.Put() | Out-Null;
        Write-Host Enabled $Id
    } else {
        Write-Error $Id is not a valid predefined key
    }
}

# Some example uses of the function defined above.

Enable-Predefined-Key "Ctrl+Alt+Delete"
Enable-Predefined-Key "Ctrl+Esc"
```

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes

WINDOWS EDITION	SUPPORTED
Windows 10 Education	Yes

## Related topics

[Keyboard Filter WMI provider reference](#)

[Keyboard Filter](#)

# WEKF\_PredefinedKey.Disable

10/2/2018 • 2 minutes to read • [Edit Online](#)

Unblocks the specified predefined key combination.

## Syntax

```
[Static] uint32 Disable(  
    [In] string PredefinedKey  
)
```

## Parameters

*PredefinedKey* [in] The predefined key combination to unblock. For a list of predefined keys, see [Predefined key combinations](#).

## Return Value

Returns an HRESULT value that indicates [WMI Non-error constant](#) or a [WMI error constant](#).

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[WEKF\\_PredefinedKey](#)

[Keyboard Filter](#)

# WEKF\_PredefinedKey.Enable

10/2/2018 • 2 minutes to read • [Edit Online](#)

This method blocks the specified predefined key combination.

## Syntax

```
[Static] uint32 Enable(  
    [In] string PredefinedKey  
)
```

## Parameters

*PredefinedKey* The predefined key combination to block. For a list of predefined keys, see [Predefined key combinations](#).

## Return Value

Returns an HRESULT value that indicates [WMI non-error constant](#) or a [WMI error constant](#).

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[WEKF\\_PredefinedKey](#)

[Keyboard Filter](#)

# WEKF\_ScanCode

10/2/2018 • 2 minutes to read • [Edit Online](#)

Blocks or unblocks key combinations by using the keyboard scan code, which is an integer number that is generated whenever a key is pressed or released.

## Syntax

```
class WEKF_ScanCode {  
    [Static] uint32 Add(  
        [In] string Modifiers,  
        [In] uint16 scancode  
    );  
    [Static] uint32 Remove(  
        [In] string Modifiers,  
        [In] uint16 Scancode  
    );  
  
    [Key] string Modifiers;  
    [Key] uint16 Scancode;  
    [Read, Write] boolean Enabled;  
}
```

## Members

The following tables list any constructors, methods, fields, and properties that belong to this class.

### Methods

METHODS	DESCRIPTION
<a href="#">WEKF_ScanCode.Add</a>	Adds a new custom scan code combination and enables Keyboard Filter to block the new scan code combination.
<a href="#">WEKF_ScanCode.Remove</a>	Removes the specified custom scan code combination. Keyboard Filter stops blocking the scan code combination that was removed.

### Properties

PROPERTY	DATA TYPE	QUALIFIERS	DESCRIPTION
<b>Modifiers</b>	string	[key]	The modifier keys that are part of the key combination to block.
<b>Scancode</b>	uint16	[key]	The scan code part of the key combination to block.

PROPERTY	DATA TYPE	QUALIFIERS	DESCRIPTION						
<b>Enabled</b>	Boolean	[read, write]	<p>Indicates whether the scan code is blocked or unblocked. This property can be one of the following values:</p> <table border="1"> <thead> <tr> <th>VALUE</th><th>DESCRIPTION</th></tr> </thead> <tbody> <tr> <td><b>true</b></td><td>Indicates that the scan code is blocked.</td></tr> <tr> <td><b>false</b></td><td>Indicates that the scan code is not blocked.</td></tr> </tbody> </table>	VALUE	DESCRIPTION	<b>true</b>	Indicates that the scan code is blocked.	<b>false</b>	Indicates that the scan code is not blocked.
VALUE	DESCRIPTION								
<b>true</b>	Indicates that the scan code is blocked.								
<b>false</b>	Indicates that the scan code is not blocked.								

### Remarks

Scan codes are generated by the keyboard whenever a key is pressed. The same physical key will always generate the same scan code, regardless of which keyboard layout is currently being used by the system.

You can specify key combinations by including the modifier keys in the *Modifiers* parameter of the **Add** method or by modifying the **Modifiers** property. The most common modifier names are "Ctrl", "Shift", "Alt", and "Win".

### Example

The following code demonstrates how to add or enable a keyboard scan code that Keyboard Filter will block by using the Windows Management Instrumentation (WMI) providers for Keyboard Filter. This example modifies the properties directly, and does not call any of the methods defined in **WEKF\_ScanCode**.

```

<#
.Synopsis
    This script shows how to use the WMI provider to enable and add
    Keyboard Filter rules through Windows Powershell on the local computer.
.Parameter ComputerName
    Optional parameter to specify a remote machine that this script should
    manage. If not specified, the script will execute all WMI operations
    locally.
#>
param (
    [String] $ComputerName
)

$CommonParams = @{"namespace"="root\standardcimv2\embedded"}
$CommonParams += $PSBoundParameters

function Enable-Scancode($Modifiers, [int]$Code) {
    <#
    .Synopsis
        Toggle on a Scancode Keyboard Filter Rule
    .Description
        Use Get-WMIObject to enumerate all WEKF_ScanCode instances,
        filter against key values of "Modifiers" and "ScanCode", and set
        that instance's "Enabled" property to 1/true.

        In the case that the ScanCode instance does not exist, add a new
        instance of WEKF_ScanCode using Set-WMIInstance.
    .Example
        Enable-Predefined-Key "Ctrl+V"

        Enable filtering of the Ctrl + V sequence.
#>

$scancode =
    Get-WMIObject -class WEKF_ScanCode @CommonParams |
        where {
            ($_.Modifiers -eq $Modifiers) -and ($_.ScanCode -eq $Code)
        }

    if($scancode) {
        $scancode.Enabled = 1
        $scancode.Put() | Out-Null
        "Enabled Custom ScanCode {0}+{1:X4}" -f $Modifiers, $Code
    } else {
        Set-WMIInstance `-
            -class WEKF_ScanCode `-
            -argument @{Modifiers="$Modifiers"; ScanCode=$Code} `-
            @CommonParams | Out-Null

        "Added Custom ScanCode {0}+{1:X4}" -f $Modifiers, $Code
    }
}

# Some example uses of the function defined above.

Enable-Scancode "Ctrl" 37

```

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No

WINDOWS EDITION	SUPPORTED
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[Keyboard Filter WMI provider reference](#)

[Keyboard Filter](#)

# WEKF\_ScanCode.Add

10/2/2018 • 2 minutes to read • [Edit Online](#)

This method adds a new custom scan code combination and enables Keyboard Filter to block the new combination.

## Syntax

```
[Static] uint32 Add(  
    [In] string Modifiers,  
    [In] uint16 ScanCode  
)
```

## Parameters

*Modifiers* The modifier keys that are part of the key combination to block.

*ScanCode* The hardware scan code of the key to block.

## Return Value

Returns an HRESULT value that indicates [WMI non-error constant](#) or a [WMI error constant](#).

## Remarks

**WEKF\_ScanCode.Add** creates a new **WEKF\_ScanCode** object and sets the **Enabled** property of the new object to **true**.

If a **WEKF\_ScanCode** object already exists with same *Modifiers* and *ScanCode* properties, then

**WEKF\_ScanCode.Add** returns an error code and does not create a new object or modify any properties of the existing object. If the existing **WEKF\_ScanCode** object has the **Enabled** property set to **false**, Keyboard Filter does not block the scan code.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[WEKF\\_ScanCode](#)

[Keyboard Filter](#)

# WEKF\_ScanCode.Remove

10/2/2018 • 2 minutes to read • [Edit Online](#)

This method removes a custom scan code key combination, causing Keyboard Filter to stop blocking the removed combination.

## Syntax

```
[Static] uint32 Remove(  
    [In] string Modifiers,  
    [In] uint16 ScanCode  
) ;
```

## Parameters

*Modifiers* The modifier keys of the combination to remove.

*ScanCode* The scan code of the combination to remove.

## Return Value

Returns an HRESULT value that indicates [WMI non-error constant](#) or a [WMI error constant](#).

## Remarks

**WEKF\_ScanCode.Remove** removes an existing **WEKF\_ScanCode** object. If the object does not exist, **WEKF\_ScanCode.Remove** returns an error with the value 0x8007007B.

Because this method is static, you cannot call it on an object instance, but must instead call it at the class level.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[WEKF\\_ScanCode](#)

[Keyboard Filter](#)

# WEKF\_Settings

10/2/2018 • 3 minutes to read • [Edit Online](#)

Enables or disables settings for Keyboard Filter.

## Syntax

```
class WEKF_Settings {  
    [Key] string Name;  
    [Read, Write] string Value;  
};
```

## Members

The following tables list any methods and properties that belong to this class.

### Properties

PROPERTY	DATA TYPE	QUALIFIERS	DESCRIPTION
<b>Name</b>	string	[key]	Indicates the name of the Keyboard Filter setting that this object represents. See the Remarks section for a list of valid setting names.
<b>Value</b>	string	[read, write]	Represents the value of the <b>Name</b> setting. The value is not case-sensitive.  See the Remarks section for a list of valid values for each setting.

### Remarks

You must be signed in to an administrator account to make any changes to this class.

Each **WEKF\_Settings** object represents a single Keyboard Filter setting. You can enumerate across all **WEKF\_Settings** objects to see the value of all Keyboard Filter settings.

The following table lists all settings available for Keyboard Filter.

SETTING NAME	DESCRIPTION
<b>DisableKeyboardFilterForAdministrators</b>	This setting specifies whether Keyboard Filter is enabled or disabled for administrator accounts. Set to <b>true</b> to disable Keyboard Filter for administrator accounts; otherwise, set to <b>false</b> . Set to <b>true</b> by default.

SETTING NAME	DESCRIPTION
<b>ForceOffAccessibility</b>	<p>This setting specifies whether Keyboard Filter blocks users from enabling Ease of Access features. Set to <b>true</b> to force disabling the Ease of Access features. Set to <b>false</b> to allow enabling the Ease of Access features. Set to <b>false</b> by default.</p> <p>Changing this setting to <b>false</b> does not automatically enable Ease of Access features; you must manually enable them.</p>
<b>BreakoutKeyScanCode</b>	<p>This setting specifies the scan code of the key that enables a user to break out of an account that is locked down with Keyboard Filter. A user can press <b>CTRL+ALT+DELETE</b> to switch to the Welcome screen.</p> <p>Set to the scan code for the left Windows logo key by default.</p>

One instance of the **WEKF\_Settings** class exists for each valid setting.

Changes to the **DisableKeyboardFilterForAdministrator** setting are applied when an administrator account signs in, and applies to all applications run during the user session. If a user without an administrator account runs an application as an administrator, Keyboard Filter is still enabled, regardless of the **DisableKeyboardFilterForAdministrator** setting.

Changes to the **BreakoutKeyScanCode** setting do not take effect until you restart the device.

If the **BreakoutKeyScanCode** is set to the scan code for either the left Windows logo key or the right Windows logo key, both Windows Logo keys will work as the breakout key.

The **BreakoutKeyScanCode** setting only applies to accounts where Keyboard Filter is active. If the scan code is set to a value that does not map to any key, such as 0 (zero), then you must use another method to access the Welcome screen if you need to service the device, such as remotely connecting, or restarting the device if automatic sign-in is not enabled.

### Important

On some devices, if the breakout key is pressed too rapidly, the key presses may not register. We recommend that you include a slight pause between each breakout key press.

#### WARNING

When setting the **BreakoutKeyScanCode**, be sure to use the scan code of the key, and not the virtual key value.

### Example

The following Windows PowerShell script demonstrates how to use this class to modify the breakout mode key for Keyboard Filter. This example sets the **BreakoutKeyScanCode** setting to the scan code for the Home key on a standard keyboard.

```

---Define variables---

$COMPUTER = "localhost"
$NAMESPACE = "root\standardcimv2\embedded"

# Define the decimal scan code of the Home key

$HomeKeyScanCode = 71

# Get the BreakoutKeyScanCode setting from WEKF_Settings

$BreakoutMode = get-wmiobject -class wekf_settings -namespace $NAMESPACE | where {$_.name -eq
"BreakoutKeyScanCode"}

# Set the breakout key to the Home key.

$BreakoutMode.value = $HomeKeyScanCode

# Push the change into the WMI configuration. You must restart your device before this change takes effect.

$BreakoutMode.put()

```

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[Keyboard Filter WMI provider reference](#)

[Keyboard Filter](#)

# Windows PowerShell script samples for Keyboard Filter

10/2/2018 • 2 minutes to read • [Edit Online](#)

The list below describes sample Windows PowerShell scripts that demonstrate how to use the Windows Management Instrumentation (WMI) providers for Keyboard Filter.

[Add blocked key combinations](#) Demonstrates how to block key combinations for Keyboard Filter.

[Disable all blocked key combinations](#) Demonstrates how to disable all blocked key combinations for Keyboard Filter.

[List all configured key combinations](#) Demonstrates how to list all defined key combination configurations for Keyboard Filter.

[Modify global settings](#) Demonstrates how to modify global settings for Keyboard Filter.

[Remove key combination configurations](#) Demonstrates how to remove a custom defined key combination configuration for Keyboard Filter.

## Related topics

[Keyboard Filter WMI provider reference](#)

[Keyboard Filter](#)

# Add blocked key combinations

10/10/2018 • 2 minutes to read • [Edit Online](#)

The following sample Windows PowerShell script uses the Windows Management Instrumentation (WMI) providers for Keyboard Filter to create three functions to configure Keyboard Filter so that Keyboard Filter blocks key combinations. It demonstrates several ways to use each function.

The first function, `Enable-PredefinedKeyFilter`, blocks key combinations that are predefined for Keyboard Filter.

The second function, `Enable-CustomKeyFilter`, blocks custom key combinations by using the English key names.

The third function, `Enable-ScancodeFilter`, blocks custom key combinations by using the keyboard scan code for the key.

## KeyFilter.psm1

```
#  
# Copyright (C) Microsoft. All rights reserved.  
#  
  
#+  
.Synopsis  
    This script shows how to use the built in WMI providers to enable and add  
    keyboard filter rules through Windows PowerShell on the local computer.  
.Parameter ComputerName  
    Optional parameter to specify a remote machine that this script should  
    manage. If not specified, the script will execute all WMI operations  
    locally.  
#+  
param (  
    [string]$ComputerName  
)  
  
$CommonParams = @{"namespace"="root\standardcimv2\embedded"}  
$CommonParams += $PSBoundParameters  
  
function Enable-PredefinedKeyFilter {  
   #+  
.Synopsis  
    Toggle on a Predefined Key keyboard filter Rule  
.Description  
    Use Get-WMIObject to enumerate all WEKF_PredefinedKey instances,  
    filter against key value "Id", and set that instance's "Enabled"  
    property to 1/true.  
.Example  
    Enable-PredefinedKeyFilter "Ctrl+Alt+Del"  
    Enable CAD filtering  
#+  
[Alias("Enable-Predefined-Key")] # The old name  
[CmdletBinding()]  
param([string]$Id)  
  
$predefined = Get-WMIObject -class WEKF_PredefinedKey @CommonParams |  
    Where-Object {  
        $_.Id -eq $Id  
    }  
  
if ($predefined) {  
    $predefined.Enabled = 1  
    $predefined.Put() | Out-Null
```

```

        Write-Host "Enabled $Id"
    } else {
        Write-Error "$Id is not a valid predefined key"
    }
}

function Enable-CustomKeyFilter {
    <#
    .Synopsis
        Toggle on a Custom Key keyboard filter Rule
    .Description
        Use Get-WMIObject to enumerate all WEKF_CustomKey instances,
        filter against key value "Id", and set that instance's "Enabled"
        property to 1/true.

        In the case that the Custom instance does not exist, add a new
        instance of WEKF_CustomKey using Set-WMIInstance.

    .Example
        Enable-CustomKeyFilter "Ctrl+V"
        Enable filtering of the Ctrl + V sequence.

    #>
    [Alias("Enable-Custom-Key")] # The old name
    [CmdletBinding()]
    param([string]$Id)

    $custom = Get-WMIObject -class WEKF_CustomKey @CommonParams |
        Where-Object {
            $_.Id -eq "$Id"
        }

    if ($custom) {
        # Rule exists. Just enable it.
        $custom.Enabled = 1
        $custom.Put() | Out-Null
        Write-Host ("Enabled Custom Filter $Id.")
    } else {
        Set-WMIInstance `

            -class WEKF_CustomKey `

            -argument @{Id="$Id"} `

            @CommonParams | Out-Null
        Write-Host ("Added Custom Filter $Id.")
    }
}

function Enable-ScancodeFilter {
    <#
    .Synopsis
        Toggle on a Scancode keyboard filter Rule
    .Description
        Use Get-WMIObject to enumerate all WEKF_Scancode instances,
        filter against key values of "Modifiers" and "Scancode", and set
        that instance's "Enabled" property to 1/true.

        In the case that the Scancode instance does not exist, add a new
        instance of WEKF_Scancode using Set-WMIInstance.

    .Example
        Enable-ScancodeFilter "Ctrl" 0x25
        Enable filtering of the Ctrl + keyboard scancode 25 (base-16)
        sequence.

    .Example
        Enable-ScancodeFilter "Ctrl" 37
        Enable filtering of the Ctrl + keyboard scancode 37 (base-10)
        sequence.

    #>
    [Alias("Enable-Scancode")] # The old name
    [CmdletBinding()]
    param([string]$Modifiers, [int]$Code)
}

```

```

$scancode =
Get-WMIObject -class WEKF_ScanCode @CommonParams |
Where-Object {
    ($_.Modifiers -eq $Modifiers) -and ($_.ScanCode -eq $Code)
}

if($scancode) {
    $scancode.Enabled = 1
    $scancode.Put() | Out-Null
    Write-Host ("Enabled Custom ScanCode {0}+{1:X4}" -f $Modifiers, $Code)
} else {
    Set-WMIInstance `

        -class WEKF_ScanCode `

        -argument @{$Modifiers="$Modifiers"; ScanCode=$Code} `

        @CommonParams | Out-Null

    Write-Host ("Added Custom ScanCode {0}+{1:X4}" -f $Modifiers, $Code)
}
}

# Some example uses of the functions defined above.
Enable-PredefinedKeyFilter "Ctrl+Alt+Del"
Enable-PredefinedKeyFilter "Ctrl+Esc"
Enable-CustomKeyFilter "Ctrl+V"
Enable-CustomKeyFilter "Numpad0"
Enable-CustomKeyFilter "Shift+Numpad1"
Enable-CustomKeyFilter "%"
Enable-ScanCodeFilter "Ctrl" 37

```

## Related topics

[Windows PowerShell script samples for keyboard filter](#)

[Keyboard filter WMI provider reference](#)

[Keyboard filter](#)

# Disable all blocked key combinations

10/2/2018 • 2 minutes to read • [Edit Online](#)

The following sample Windows PowerShell script uses the WMI providers to disable all blocked key combinations for Keyboard Filter by using the Windows Management Instrumentation (WMI) providers for Keyboard Filter. The key combination configurations are not removed, but Keyboard Filter stops blocking any keys.

## Disable-all-rules.ps1

```

#
# Copyright (C) Microsoft. All rights reserved.
#

<#
.Synopsis
    This Windows PowerShell script shows how to enumerate all existing keyboard filter
    rules and how to disable them by setting the Enabled property directly.
.Description
    For each instance of WEKF_PredefinedKey, WEKF_CustomKey, and WEKF_ScanCode,
    set the Enabled property to false/0 to disable the filter rule, thus
    allowing all key sequences through the filter.
.Parameter ComputerName
    Optional parameter to specify the remote computer that this script should
    manage. If not specified, the script will execute all WMI operations
    locally.
#>

param(
    [String]$ComputerName
)

$CommonParams = @{"namespace"="root\standardcimv2\embedded"}
$CommonParams += $PSBoundParameters

Get-WMIObject -class WEKF_PredefinedKey @CommonParams |
    foreach {
        if ($_.Enabled) {
            $_.Enabled = 0;
            $_.Put() | Out-Null;
            Write-Host Disabled $_.Id
        }
    }

Get-WMIObject -class WEKF_CustomKey @CommonParams |
    foreach {
        if ($_.Enabled) {
            $_.Enabled = 0;
            $_.Put() | Out-Null;
            Write-Host Disabled $_.Id
        }
    }

Get-WMIObject -class WEKF_ScanCode @CommonParams |
    foreach {
        if ($_.Enabled) {
            $_.Enabled = 0;
            $_.Put() | Out-Null;
            "Disabled {0}+{1:X4}" -f $_.Modifiers,$_.ScanCode
        }
    }

```

## Related topics

[Windows PowerShell script samples for keyboard filter](#)

[Keyboard filter WMI provider reference](#)

[Keyboard filter](#)

# List all configured key combinations

10/2/2018 • 2 minutes to read • [Edit Online](#)

The following sample Windows PowerShell script uses the Windows Management Instrumentation (WMI) providers for Keyboard Filter to displays all key combination configurations for Keyboard Filter.

## List-rules.ps1

```
#  
# Copyright (C) Microsoft. All rights reserved.  
#  
  
<#  
.Synopsis  
    Enumerate all active keyboard filter rules on the system.  
.Description  
    For each instance of WEKF_PredefinedKey, WEKF_CustomKey, and WEKF_Scancode,  
    get the Enabled property. If Enabled, then output a short description  
    of the rule.  
.Parameter ComputerName  
    Optional parameter to specify the remote machine that this script should  
    manage. If not specified, the script will execute all WMI operations  
    locally.  
#>  
param (  
    [String] $ComputerName  
)  
  
$CommonParams = @{"namespace"="root\standardcimv2\embedded"}  
$CommonParams += $PSBoundParameters  
  
write-host Enabled Predefined Keys -foregroundcolor cyan  
Get-WMIObject -class WEKF_PredefinedKey @CommonParams |  
    foreach {  
        if ($_.Enabled) {  
            write-host $_.Id  
        }  
    }  
  
write-host Enabled Custom Keys -foregroundcolor cyan  
Get-WMIObject -class WEKF_CustomKey @CommonParams |  
    foreach {  
        if ($_.Enabled) {  
            write-host $_.Id  
        }  
    }  
  
write-host Enabled Scancodes -foregroundcolor cyan  
Get-WMIObject -class WEKF_Scancode @CommonParams |  
    foreach {  
        if ($_.Enabled) {  
            "{0}+{1:X4}" -f $_.Modifiers, $_.Scancode  
        }  
    }
```

## Related topics

[Windows PowerShell script samples for keyboard filter](#)

## Keyboard filter WMI provider reference

### Keyboard filter

# Modify global settings

10/2/2018 • 2 minutes to read • [Edit Online](#)

The following sample Windows PowerShell scripts use the Windows Management Instrumentation (WMI) providers to modify global settings for Keyboard Filter.

The function **Get-Setting** retrieves the value of a global setting for Keyboard Filter.

In the first script, the function **Set-DisableKeyboardFilterForAdministrators** modifies the value of the **DisableKeyboardFilterForAdministrators** setting.

In the second script, the function **Set-ForceOffAccessibility** modifies the value of the **ForceOffAccessibility** setting.

## Set-DisableKeyboardFilterForAdministrators.ps1

```
#  
# Copyright (C) Microsoft. All rights reserved.  
#  
  
<#  
.Synopsis  
    This script shows how to enumerate WEKF_Settings to find global settings  
    that can be set on the keyboard filter. In this specific script, the  
    global setting to be set is "DisableKeyboardFilterForAdministrators".  
.Parameter ComputerName  
    Optional parameter to specify a remote computer that this script should  
    manage. If not specified, the script will execute all WMI operations  
    locally.  
.Parameter On  
    Switch if present that sets "DisableKeyboardFilterForAdministrators" to  
    true. If not present, sets the setting to false.  
#>  
  
param (  
    [Switch] $On = $False,  
    [String] $ComputerName  
)  
  
$CommonParams = @{"namespace"="root\standardcimv2\embedded"};  
if ($PSBoundParameters.ContainsKey("ComputerName")) {  
    $CommonParams += @{"ComputerName" = $ComputerName};  
}  
  
function Get-Setting([String] $Name) {  
    <#  
.Synopsis  
    Get a WMIOBJECT by name from WEKF_Settings  
.Parameter Name  
    The name of the setting, which is the key for the WEKF_Settings class.  
#>  
    $Entry = Get-WMIOBJECT -class WEKF_Settings @CommonParams |  
        where {  
            $_.Name -eq $Name  
        }  
  
    return $Entry  
}  
  
function Set-DisableKeyboardFilterForAdministrators([Bool] $Value) {
```

```

<#
.Synopsis
    Set the DisableKeyboardFilterForAdministrators setting to true or
    false.
.Description
    Set DisableKeyboardFilterForAdministrators to true or false based
    on $Value
.Parameter Value
    A Boolean value
#>

$Setting = Get-Setting("DisableKeyboardFilterForAdministrators")
if ($Setting) {
    if ($Value) {
        $Setting.Value = "true"
    } else {
        $Setting.Value = "false"
    }
    $Setting.Put() | Out-Null;
} else {
    Write-Error "Unable to find DisableKeyboardFilterForAdministrators setting";
}
}

Set-DisableKeyboardFilterForAdministrators $On

```

## Set-ForceOffAccessibility.ps1

```

#
# Copyright (C) Microsoft. All rights reserved.
#

<#
.Synopsis
    This script shows how to enumerate WEKF_Settings to find global settings
    that can be set on the keyboard filter. In this specific script, the
    global setting to be set is "ForceOffAccessibility".
.Parameter ComputerName
    Optional parameter to specify a remote computer that this script should
    manage. If not specified, the script will execute all WMI operations
    locally.
.Parameter Enabled
    Switch if present that sets "ForceOffAccessibility" to true. If not
    present, sets the setting to false.
#>

param (
    [Switch] $Enabled = $False,
    [String] $ComputerName
)

$CommonParams = @{"namespace"="root\standardcimv2\embedded"};
if ($PSBoundParameters.ContainsKey("ComputerName")) {
    $CommonParams += @{$"ComputerName" = $ComputerName};
}

function Get-Setting([String] $Name) {
    <#
.Synopsis
    Get a WMIOObject by name from WEKF_Settings
.Parameter Name
    The name of the setting, which is the key for the WEKF_Settings class.
#>
    $Entry = Get-WMIOObject -class WEKF_Settings @CommonParams |
        where {
            $_.Name -eq $Name

```

```
}

return $Entry
}

function Set-ForceOffAccessibility([Bool] $Value) {
<#
.Synopsis
    Set the ForceOffAccessibility setting to true or false.
.Description
    Set ForceOffAccessibility to true or false based on $Value
.Parameter Value
    A Boolean value
#>

$Setting = Get-Setting("ForceOffAccessibility")
if ($Setting) {
    if ($Value) {
        $Setting.Value = "true"
    } else {
        $Setting.Value = "false"
    }
    $Setting.Put() | Out-Null;
} else {
    Write-Error "Unable to find ForceOffAccessibility setting";
}
}

Set-ForceOffAccessibility $Enabled
```

## Related topics

[Windows PowerShell script samples for keyboard filter](#)

[WEKF\\_Settings](#)

[Keyboard filter](#)

# Remove key combination configurations

10/2/2018 • 2 minutes to read • [Edit Online](#)

The following sample Windows PowerShell script uses the Windows Management Instrumentation (WMI) providers for Keyboard Filter to create two functions to remove custom-defined key combination configurations from Keyboard Filter. It demonstrates several ways to use each function.

The first function, **Remove-Custom-Key**, removes custom key combination configurations.

The second function, **Remove-Scancode**, removes custom scan code configurations.

You cannot remove the predefined key combination configurations for Keyboard Filter, but you can disable them.

## Remove-rules.ps1

```
#  
# Copyright (C) Microsoft. All rights reserved.  
#  
  
#+  
.Synopsis  
    This script shows how to use the build in WMI providers to remove keyboard filter rules. Rules of type  
WEKF_PredefinedKey cannot be removed.  
.Parameter ComputerName  
    Optional parameter to specify the remote computer that this script should  
manage. If not specified, the script will execute all WMI operations  
locally.  
#+  
  
param(  
    [string] $ComputerName  
)  
  
$CommonParams = @{"namespace"="root\standardcimv2\embedded"}  
$CommonParams += $PSBoundParameters  
  
function Remove-Custom-Key($Id) {  
   #+  
.Synopsis  
    Remove an instance of WEKF_CustomKey  
.Description  
    Enumerate all instances of WEKF_CustomKey. When an instance has an  
Id that matches $Id, delete it.  
.Example  
    Remove-Custom-Key "Ctrl+V"  
  
    This removes the instance of WEKF_CustomKey with a key Id of "Ctrl+V"  
#+  
  
    $customInstance = Get-WMIObject -class WEKF_CustomKey @CommonParams |  
        where {$_.Id -eq $Id}  
  
    if ($customInstance) {  
        $customInstance.Delete();  
        "Removed Custom Filter $Id.";  
    } else {  
        "Custom Filter $Id does not exist.";  
    }  
}
```

```

function Remove-Scancode($Modifiers, [int]$Code) {
    <#
    .Synopsis
        Remove an instance of WEKF_Scancode
    .Description
        Enumerate all instances of WEKF_Scancode. When an instance has a
        matching modifiers and code, delete it.
    .Example
        Remove-Scancode "Ctrl" 37

        This removes the instance of WEKF_Scancode with Modifiers="Ctrl" and
        Scancode=37.

#>

$scancodeInstance = Get-WMIObject -class WEKF_Scancode @CommonParams |
    where {($_.Modifiers -eq $Modifiers) -and ($_.Scancode -eq $Code)}

if ($scancodeInstance) {
    $scancodeInstance.Delete();
    "Removed Scancode $Modifiers+$Code.";
} else {
    "Scancode $Modifiers+$Code does not exist.";
}
}

# Some example uses of the functions defined above.
Remove-Custom-Key "Ctrl+V"
Remove-Custom-Key "Numpad0"
Remove-Custom-Key "Shift+Numpad1"
Remove-Custom-Key "%"
Remove-Scancode "Ctrl" 37

```

## Related topics

[Windows PowerShell script samples for keyboard filter](#)

[Keyboard filter WMI provider reference](#)

[Keyboard filter](#)

# Shell Launcher

10/2/2018 • 10 minutes to read • [Edit Online](#)

You can use Shell Launcher to replace the default Windows 10 shell with a custom shell. You can use almost any application or executable as your custom shell, such as a command window or a custom dedicated application.

You can also configure Shell Launcher to launch different shell applications for different users or user groups.

There are a few exceptions to the applications and executables you can use as a custom shell:

- You cannot use the following executable as a custom shell: `c:\\Windows\\\\System32\\\\Eshell.exe`. Using Eshell.exe as the default shell will result in a blank screen after user signs in.
- You cannot use a Universal Windows app as a custom shell.
- You cannot use a custom shell to launch Universal Windows apps, for example, the Settings app.
- You cannot use an application that launches a different process and exits as a custom shell. For example, you cannot specify **write.exe** in Shell Launcher. Shell Launcher launches a custom shell and monitors the process to identify when the custom shell exits. **Write.exe** creates a 32-bit wordpad.exe process and exits. Because Shell Launcher is not aware of the newly created wordpad.exe process, Shell Launcher will take action based on the exit code of **Write.exe**, and restart the custom shell.

## NOTE

You cannot configure both Shell Launcher and assigned access on the same system.

Shell Launcher processes the **Run** and **RunOnce** registry keys before starting the custom shell, so your custom shell doesn't need to handle the automatic startup of other applications and services.

Shell Launcher also handles the behavior of the system when your custom shell exits. You can configure the shell exit behavior if the default behavior does not meet your needs.

## Requirements

Windows 10 Enterprise or Windows 10 Education.

## Terminology

- **Turn on, enable:** To make the setting available to the device and optionally apply the settings to the device.
- **Configure:** To customize the setting or sub-settings.
- **Embedded Shell Launcher:** This feature is called Embedded Shell Launcher in Windows 10, version 1511.
- **Custom Shell Launcher:** This feature is called Shell Launcher in Windows 10, version 1607 and later.

## Turn on Shell Launcher

Shell Launcher is an optional component and is not turned on by default in Windows 10. It must be turned on prior to configuring. You can turn on and configure Shell Launcher in a customized Windows 10 image (.wim) if Microsoft Windows has not been installed. If Windows has already been installed and you are applying a provisioning package to configure Shell Launcher, you must first turn on Shell Launcher in order for a provisioning package to successfully apply.

### Enable Shell Launcher using Control Panel

1. In the **Search the web and Windows** field, type **Programs and Features** and either press **Enter** or tap or click **Programs and Features** to open it.
2. In the **Programs and Features** window, click **Turn Windows features on or off**.
3. In the **Windows Features** window, expand the **Device Lockdown** node, select or clear the checkbox for **Shell Launcher**, and then click **OK**.
4. The **Windows Features** window indicates that Windows is searching for required files and displays a progress bar. Once found, the window indicates that Windows is applying the changes. When completed, the window indicates the requested changes are completed.
5. Click **Close** to close the **Windows Features** window.

**NOTE**

Turning on Shell Launcher does not require a device restart.

### Enable Shell Launcher by calling WESL\_UserSetting

1. Enable or disable Shell Launcher by calling the `WESL_UserSetting.SetEnabled` function in the Windows Management Instrumentation (WMI) class `WESL_UserSetting`.
2. If you enable or disable Shell Launcher using `WESL_UserSetting`, the changes do not affect any sessions that are currently signed in; you must sign out and sign back in.

This example uses a Windows image called `install.wim`, but you can use the same procedure to apply a provisioning package (for more information on DISM, see [What Is Deployment Image Servicing and Management](#)).

### Enable Shell Launcher using DISM

1. Open a command prompt with administrator privileges.
2. Copy `install.wim` to a temporary folder on hard drive (in the following steps, we'll assume it's called `C:\wim`).
3. Create a new directory.

```
md c:\wim
```

4. Mount the image.

```
dism /mount-wim /wimfile:c:\bootmedia\sources\install.wim /index:1 /MountDir:c:\wim
```

5. Enable the feature.

```
dism /image:c:\wim /enable-feature /all /featureName:Client-EmbeddedShellLauncher
```

6. Commit the change.

```
dism /unmount-wim /MountDir:c:\wim /Commit
```

### Enable Shell Launcher using Windows Configuration Designer

The Shell Launcher settings are also available as Windows provisioning settings so you can configure these settings to be applied during the image runtime. You can set one or all Shell Launcher settings by creating a provisioning package using Windows Configuration Designer and then applying the provisioning package during image deployment time or runtime. If Windows has not been installed and you are using Windows Configuration Designer to create installation media with settings for Shell Launcher included in the image or you

are applying a provisioning package during setup, you must enable Shell Launcher on the installation media with DISM in order for a provisioning package to successfully apply.

Use the following steps to create a provisioning package that contains the ShellLauncher settings.

1. Build a provisioning package in Windows Configuration Designer by following the instructions in [Create a provisioning package for Windows 10](#).
2. In the **Available customizations** page, select **Runtime settings > SMI Settings > ShellLauncher**.
3. Set the value of **Enable** to **ENABLE**. Additional options to configure Shell Launcher will appear, and you can set the values as desired.
4. Once you have finished configuring the settings and creating the provisioning package, you can apply the package to the image deployment time or runtime. See the [Apply a provisioning package](#) for more information. Note that the process for applying the package to a Windows 10 Enterprise image is the same.

## Configure Shell Launcher

There are two ways you can configure Shell Launcher:

1. In Windows 10, version 1803, you can configure Shell Launcher using the **ShellLauncher** node of the Assigned Access Configuration Service Provider (CSP). See [AssignedAccess CSP](#) for details. Configuring Shell Launcher using this method also automatically enables Shell Launcher on the device, if the device supports it.
2. Use the Shell Launcher WMI providers directly in a PowerShell script or application.

You can configure the following options for Shell Launcher:

- Enable or disable Shell Launcher.
- Specify a shell configuration for a specific user or group.
- Remove a shell configuration for a specific user or group.
- Change the default shell configuration.
- Get information on a shell configuration for a specific user or group.

Any changes do not take effect until a user signs in.

## Launch different shells for different user accounts

By default, Shell Launcher runs the default shell, which is specified when you create the OS image at design time. The default shell is set to Cmd.exe, but you can specify any executable file to be the default shell.

You can configure Shell Launcher to launch a different shell for specific users or groups if you do not want to run the default shell. For example, you might configure a device to run a custom application shell for guest accounts, but run the standard Windows Explorer shell for administrator accounts in order to service the device.

If you use the WMI providers to configure Shell Launcher for a user or group at run time, you must use the security identifier (SID) for that user or group; you cannot use the user name or group name.

For more information about common security identifiers, see [Well-known SIDs](#).

When the current signed in account belongs to two or more groups that have different configurations defined for each group, Shell Launcher uses the first configuration it finds. The search order is not defined, so we recommend that you avoid assigning a user to multiple groups with different Shell Launcher configurations.

## Perform an action when the shell exits

When a custom shell exits, Shell Launcher can perform one of four actions, based on the following return codes:

RETURN CODE	ACTION
0	Restart the shell.
1	Restart the device.
2	Shut down the device.
3	Do nothing.

#### IMPORTANT

Make sure that your shell application does not automatically exit and is not automatically closed by any features such as Dialog Filter, as this can lead to an infinite cycle of exiting and restarting, unless the return code action is set to do nothing.

#### Default return code action

You can define a default return code action for Shell Launcher with the DefaultReturnCodeAction setting. If you do not change the initial value, the default return code action is set to 0 (zero), which indicates that Shell Launcher restarts the shell when the shell exits.

#### Map the exit code to a Shell Launcher action

Shell Launcher can take a specific action based on the exit code returned by the shell. For any given exit code returned by the shell, you can configure the action that Shell Launcher takes by mapping that exit code to one of the shell exit actions.

If the exit code does not match a defined value, Shell Launcher performs the default return code action.

For example, your shell might return exit code values of -1, 0, or 255 depending on how the shell exits. You can configure Shell Launcher to restart the system (1) when the shell returns a value of -1, restart the shell (0) when the shell returns a value of 0, and shut down the system (2) when the shell returns a value of 255. Your custom return code action mapping would look like this:

EXIT CODE	ACTION
-1	1 (restart the system)
0	0 (restart the app)
255	2 (shut down the system)

## Set your custom shell

Modify the following PowerShell script as appropriate and run the script on the device.

```
# Check if shell launcher license is enabled
function Check-ShellLauncherLicenseEnabled
{
    [string]$source = @"
using System;
```

```

using System.Runtime.InteropServices;

static class CheckShellLauncherLicense
{
    const int S_OK = 0;

    public static bool IsShellLauncherLicenseEnabled()
    {
        int enabled = 0;

        if (NativeMethods.SLGetWindowsInformationDWORD("EmbeddedFeature-ShellLauncher-Enabled", out enabled) != S_OK)
        {
            enabled = 0;
        }
        return (enabled != 0);
    }

    static class NativeMethods
    {
        [DllImport("Slc.dll")]
        internal static extern int SLGetWindowsInformationDWORD([MarshalAs(UnmanagedType.LPWStr)]string valueName, out int value);
    }
}

}

"@

$type = Add-Type -TypeDefinition $source -PassThru

return $type[0]::IsShellLauncherLicenseEnabled()
}

[bool]$result = $false

$result = Check-ShellLauncherLicenseEnabled
```nShell Launcher license enabled is set to " + $result
if (-not($result))
{
    ```nThis device doesn't have required license to use Shell Launcher"
    exit
}

$COMPUTER = "localhost"
$NAMESPACE = "root\standardcimv2\embedded"

# Create a handle to the class instance so we can call the static methods.
try {
    $ShellLauncherClass = [wmiclass]"\\$COMPUTER\${$NAMESPACE}:WESL_UserSetting"
} catch [Exception] {
    write-host $_.Exception.Message;
    write-host "Make sure Shell Launcher feature is enabled"
    exit
}

# This well-known security identifier (SID) corresponds to the BUILTIN\Administrators group.

$Admins_SID = "S-1-5-32-544"

# Create a function to retrieve the SID for a user account on a machine.

function Get-UsernameSID($AccountName) {

    $NTUserObject = New-Object System.Security.Principal.NTAccount($AccountName)
    $NTUserSID = $NTUserObject.Translate([System.Security.Principal.SecurityIdentifier])

    return $NTUserSID.Value
}

```

```

# Get the SID for a user account named "Cashier". Rename "Cashier" to an existing account on your system to
test this script.

$Cashier_SID = Get-UsernameSID("Cashier")

# Define actions to take when the shell program exits.

$restart_shell = 0
$restart_device = 1
$shutdown_device = 2

# Examples. You can change these examples to use the program that you want to use as the shell.

# This example sets the command prompt as the default shell, and restarts the device if the command prompt is
closed.

$ShellLauncherClass.SetDefaultShell("cmd.exe", $restart_device)

# Display the default shell to verify that it was added correctly.

$DefaultShellObject = $ShellLauncherClass.GetDefaultShell()

```
nDefault Shell is set to " + $DefaultShellObject.Shell + " and the default action is set to " +
$DefaultShellObject.defaultaction

# Set Internet Explorer as the shell for "Cashier", and restart the machine if Internet Explorer is closed.

$ShellLauncherClass.SetCustomShell($Cashier_SID, "c:\program files\internet explorer\iexplore.exe
www.microsoft.com", ($null), ($null), $restart_shell)

# Set Explorer as the shell for administrators.

$ShellLauncherClass.SetCustomShell($Admins_SID, "explorer.exe")

# View all the custom shells defined.

```
nCurrent settings for custom shells:"+
Get-WmiObject -namespace $NAMESPACE -computer $COMPUTER -class WESL_UserSetting | Select Sid, Shell,
DefaultAction

# Enable Shell Launcher

$ShellLauncherClass.SetEnabled($TRUE)

$IsShellLauncherEnabled = $ShellLauncherClass.IsEnabled()

```
nEnabled is set to " + $IsShellLauncherEnabled.Enabled

# Remove the new custom shells.

$ShellLauncherClass.RemoveCustomShell($Admins_SID)

$ShellLauncherClass.RemoveCustomShell($Cashier_SID)

# Disable Shell Launcher

$ShellLauncherClass.SetEnabled($FALSE)

$IsShellLauncherEnabled = $ShellLauncherClass.IsEnabled()

```
nEnabled is set to " + $IsShellLauncherEnabled.Enabled

```

**NOTE**

The script above includes examples of multiple configuration options, including removing a custom shell and disabling Shell Launcher. It is not intended to be run as-is.

## Shell Launcher user rights

A custom shell is launched with the same level of user rights as the account that is signed in. This means that a user with administrator rights can perform any system action that requires administrator rights, including launching other applications with administrator rights, while a user without administrator rights cannot.

**WARNING**

If your shell application requires administrator rights and needs to be elevated, and User Account Control (UAC) is present on your device, you must disable UAC in order for Shell Launcher to launch the shell application.

## Related topics

[Unbranded Boot](#)

[Custom Logon](#)

# WESL\_UserSetting

10/2/2018 • 4 minutes to read • [Edit Online](#)

This class configures which application Shell Launcher starts based on the security identifier (SID) of the signed in user, and also configures the set of return codes and return actions that Shell Launcher performs when the application exits.

## Syntax

```
class WESL_UserSetting {
    [read, write, Required] string Sid;
    [read, write, Required] string Shell;
    [read, write] Sint32 CustomReturnCodes[];
    [read, write] Sint32 CustomReturnCodesAction[];
    [read, write] sint32 DefaultAction;

    [Static] uint32 SetCustomShell(
        [In, Required] string Sid,
        [In, Required] string Shell,
        [In] sint32 CustomReturnCodes[],
        [In] sint32 CustomReturnCodesAction[],
        [In] sint32 DefaultAction
    );
    [Static] uint32 GetCustomShell(
        [In, Required] string Sid,
        [Out, Required] string Shell,
        [Out, Required] sint32 CustomReturnCodes[],
        [Out, Required] sint32 CustomReturnCodesAction[],
        [Out, Required] sint32 DefaultAction
    );
    [Static] uint32 RemoveCustomShell(
        [In, Required] string Sid
    );
    [Static] uint32 GetDefaultShell(
        [Out, Required] string Shell,
        [Out, Required] sint32 DefaultAction
    );
    [Static] uint32 SetDefaultShell(
        [In, Required] string Shell,
        [In, Required] sint32 DefaultAction
    );
    [Static] uint32 IsEnabled(
        [Out, Required] boolean Enabled
    );
    [Static] uint32 SetEnabled(
        [In, Required] boolean Enabled);
}
```

## Members

The following tables list any methods and properties that belong to this class.

### Methods

METHODS	DESCRIPTION
<a href="#">WESL_UserSetting.SetCustomShell</a>	Configures Shell Launcher for a specific user or group, based on SID.
<a href="#">WESL_UserSetting.GetCustomShell</a>	Retrieves the Shell Launcher configuration for a specific user or group, based on the SID.
<a href="#">WESL_UserSetting.RemoveCustomShell</a>	Removes a Shell Launcher configuration for a specific user or group, based on the SID.
<a href="#">WESL_UserSetting.GetDefaultShell</a>	Retrieves the default Shell Launcher configuration.
<a href="#">WESL_UserSetting.SetDefaultShell</a>	Sets the default Shell Launcher configuration.
<a href="#">WESL_UserSetting.IsEnabled</a>	Retrieves a value that indicates if Shell Launcher is enabled or disabled.
<a href="#">WESL_UserSetting.SetEnabled</a>	Enables or disables Shell Launcher.

## Properties

PROPERTY	DATA TYPE	QUALIFIERS	DESCRIPTION
<b>Sid</b>	string	[read, write, required]	User or group SID.
<b>shell</b>	string	[read, write, required]	<p>The application to start as the shell.</p> <p>The <b>shell</b> property can be a filename in the <i>Path</i> environment variable, or it can contain a fully qualified path to the application. You can also use environment variables in the path.</p> <p>Any spaces in the <b>shell</b> property must be part of a quote-delimited string.</p>
<b>CustomReturnCodes</b>	Sint32[]	[read, write]	An array of custom return codes that can be returned by the shell.

PROPERTY	DATA TYPE	QUALIFIERS	DESCRIPTION								
<b>CustomReturnCodesAction</b>	Sint32[]	[read, write]	<p>An array of custom return code actions that determine what action Shell Launcher takes when the shell exits. The custom actions map to the array of <b>CustomReturnCodes</b>.</p> <p>The possible actions are defined in the following table:</p> <table border="1"> <thead> <tr> <th>VALUE</th><th>DESCRIPTION</th></tr> </thead> <tbody> <tr> <td>0</td><td>Restarts the shell.</td></tr> <tr> <td>1</td><td>Restarts the device</td></tr> <tr> <td>2</td><td>Shuts down the device</td></tr> </tbody> </table>	VALUE	DESCRIPTION	0	Restarts the shell.	1	Restarts the device	2	Shuts down the device
VALUE	DESCRIPTION										
0	Restarts the shell.										
1	Restarts the device										
2	Shuts down the device										
<b>DefaultAction</b>	Sint32	[read, write]	<p>The default action Shell Launcher takes when the shell exits.</p> <p>The possible actions are defined in the following table:</p> <table border="1"> <thead> <tr> <th>VALUE</th><th>DESCRIPTION</th></tr> </thead> <tbody> <tr> <td>0</td><td>Restarts the shell.</td></tr> <tr> <td>1</td><td>Restarts the device</td></tr> <tr> <td>2</td><td>Shuts down the device</td></tr> </tbody> </table>	VALUE	DESCRIPTION	0	Restarts the shell.	1	Restarts the device	2	Shuts down the device
VALUE	DESCRIPTION										
0	Restarts the shell.										
1	Restarts the device										
2	Shuts down the device										

## Remarks

Only one **WESL\_UserSetting** instance exists on a device with Shell Launcher.

Shell Launcher uses the custom configuration defined for the SID of the user currently signed in, if one exists. Otherwise, Shell Launcher uses a custom configuration defined for a group SID that the user is a member of, if any exist. If multiple group custom configurations for the user exist, Shell Launcher uses the first valid configuration it finds. The search order is not defined.

If there is no custom configuration for the user's SID or any group SIDs that the user is a member of, Shell Launcher uses the default configuration.

You can find the SID for a user and any groups that the user is a member of by using the [whoami](#) command-line tool.

## Example

The following Windows PowerShell script demonstrates how to add and remove custom shell configurations for Shell Launcher by using the Windows Management Instrumentation (WMI) providers for Shell Launcher.

```

$COMPUTER = "localhost"
$NAMESPACE = "root\standardcimv2\embedded"

# Create a handle to the class instance so we can call the static methods.
$ShellLauncherClass = [wmiclass]"\\\$COMPUTER\$\\$NAMESPACE":WESL_UserSetting"

# This well-known security identifier (SID) corresponds to the BUILTIN\Administrators group.

$Admins_SID = "S-1-5-32-544"

# Create a function to retrieve the SID for a user account on a machine.

function Get-UsernameSID($AccountName) {

    $NTUserObject = New-Object System.Security.Principal.NTAccount($AccountName)
    $NTUserSID = $NTUserObject.Translate([System.Security.Principal.SecurityIdentifier])

    return $NTUserSID.Value
}

# Get the SID for a user account named "Cashier". Rename "Cashier" to an existing account on your system to
# test this script.

$Cashier_SID = Get-UsernameSID("Cashier")

# Define actions to take when the shell program exits.

$restart_shell = 0
$restart_device = 1
$shutdown_device = 2

# Examples

# Set the command prompt as the default shell, and restart the device if it's closed.

$ShellLauncherClass.SetDefaultShell("cmd.exe", $restart_device)

# Display the default shell to verify that it was added correctly.

$DefaultShellObject = $ShellLauncherClass.GetDefaultShell()

```
Default Shell is set to " + $DefaultShellObject.Shell + " and the default action is set to " +
$DefaultShellObject.defaultaction
```

# Set Internet Explorer as the shell for "Cashier", and restart the machine if it's closed.

$ShellLauncherClass.SetCustomShell($Cashier_SID, "c:\program files\internet explorer\iexplore.exe
www.microsoft.com", ($null), ($null), $restart_shell)

# Set Explorer as the shell for administrators.

$ShellLauncherClass.SetCustomShell($Admins_SID, "explorer.exe")

# View all the custom shells defined.

```
nCurrent settings for custom shells:
Get-WmiObject -namespace $NAMESPACE -computer $COMPUTER -class WESL_UserSetting | Select Sid, Shell,
DefaultAction
```

# Remove the new custom shells.

$ShellLauncherClass.RemoveCustomShell($Admins_SID)

$ShellLauncherClass.RemoveCustomShell($Cashier_SID)

```

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[Shell Launcher](#)

# WESL\_UserSetting.GetCustomShell

10/2/2018 • 2 minutes to read • [Edit Online](#)

This method retrieves the Shell Launcher configuration for a specific user or group, based on the security identifier (SID).

## Syntax

```
[Static] uint32 GetCustomShell (
    [In, Required] string Sid,
    [Out, Required] string Shell,
    [Out, Required] sint32 CustomReturnCodes[],
    [Out, Required] sint32 CustomReturnCodesAction[],
    [Out, Required] sint32 DefaultAction
);
```

## Parameters

*Sid* [in, required] A string containing the security identifier (SID) of the user or group that Shell Launcher is configured for.

*Shell* [out, required] The application or executable that Shell Launcher starts as the shell.

*CustomReturnCodes* [out, required] An array of custom return codes returned by the shell application.

*CustomReturnCodesAction* [out, required] An array of custom return code actions that determine the action that Shell Launcher takes when the shell application exits. The custom actions map to the array of *CustomReturnCodes*.

The possible actions are defined in the following table:

VALUE	DESCRIPTION
0	Restarts the shell application.
1	Restarts the device.
2	Turns off the device.

*DefaultAction* [out, required] The default action that Shell Launcher takes when the shell application exits.

The possible actions are defined in the following table:

VALUE	DESCRIPTION
0	Restarts the shell application.
1	Restarts the device.

VALUE	DESCRIPTION
2	Turns off the device.

## Return Value

Returns an HRESULT value that indicates [WMI status](#) or a [WMI error](#).

## Remarks

Shell Launcher uses the *CustomReturnCodes* and *CustomReturnCodesAction* arrays to determine the system behavior when the shell application exits, based on the return value of the application.

If the return value does not exist in *CustomReturnCodes*, or if the corresponding action defined in *CustomReturnCodesAction* is not a valid value, Shell Launcher uses *DefaultAction* to determine system behavior. If *DefaultAction* is not defined, or is not a valid value, Shell Launcher restarts the shell application.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[WESL\\_UserSetting](#)

[Shell Launcher](#)

# WESL\_UserSetting.RemoveCustomShell

10/2/2018 • 2 minutes to read • [Edit Online](#)

This method removes a Shell Launcher configuration for a specific user or group, based on the security identifier (SID).

## Syntax

```
[Static] uint32 RemoveCustomShell (
    [In, Required] string Sid
);
```

## Parameters

### Sid

[in, required] A string containing the security identifier (SID) of the user or group that Shell Launcher is configured for.

## Return Value

Returns an HRESULT value that indicates [WMI status](#) or a [WMI error](#).

## Remarks

You must restart your device for the changes to take effect.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[WESL\\_UserSetting](#)

[Shell Launcher](#)

# WESL\_UserSetting.SetCustomShell

10/2/2018 • 2 minutes to read • [Edit Online](#)

This method configures Shell Launcher for a specific user or group, based on the security identifier (SID).

## Syntax

```
[Static] uint32 SetCustomShell (
    [In, Required] string Sid,
    [In, Required] string Shell,
    [In] sint32 CustomReturnCodes[],
    [In] sint32 CustomReturnCodesAction[],
    [In] sint32 DefaultAction
);
```

## Parameters

*Sid* [in, required] A string containing the security identifier (SID) of the user or group that Shell Launcher is being configured for.

*Shell* [in, required] The application or executable that Shell Launcher starts as the shell.

*CustomReturnCodes* [in] An array of custom return codes that can be returned by the shell application.

*CustomReturnCodesAction* [in] An array of custom return code actions that determine the action that Shell Launcher takes when the shell application exits. The custom actions map to the array of *CustomReturnCodes*.

The possible actions are defined in the following table:

VALUE	DESCRIPTION
0	Restarts the shell application.
1	Restarts the device.
2	Shuts down the device.

*DefaultAction* [In] The default action that Shell Launcher takes when the shell application exits.

The possible actions are defined in the following table:

VALUE	DESCRIPTION
0	Restarts the shell application.
1	Restarts the device.

VALUE	DESCRIPTION
2	Shuts down the device.

## Return Value

Returns an HRESULT value that indicates [WMI status](#) or a [WMI error](#).

## Remarks

Shell Launcher uses the *CustomReturnCodes* and *CustomReturnCodesAction* arrays to determine the system behavior when the shell application exits, based on the return value of the shell application.

If the return value does not exist in *CustomReturnCodes*, or if the corresponding action defined in *CustomReturnCodesAction* is not a valid value, Shell Launcher uses *DefaultAction* to determine system behavior. If *DefaultAction* is not defined, or is not a valid value, Shell Launcher restarts the shell application.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[WESL\\_UserSetting](#)

[Shell Launcher](#)

# WESL\_UserSetting.SetDefaultShell

10/2/2018 • 2 minutes to read • [Edit Online](#)

This method sets the default Shell Launcher configuration.

## Syntax

```
[Static] uint32 SetDefaultShell (
    [In, Required] string Shell,
    [In, Required] sint32 DefaultAction
);
```

## Parameters

*Shell* [in, required] The application or executable that Shell Launcher starts as the shell.

*DefaultAction* [in, required] The default action that Shell Launcher takes when the *Shell* application exits.

The possible actions are defined in the following table:

VALUE	DESCRIPTION
0	Restarts the shell application.
1	Restarts the device.
2	Shuts down the device.

## Return Value

Returns an HRESULT value that indicates [WMI status](#) or a [WMI error](#).

## Remarks

Shell Launcher uses the default configuration when the security identifier (SID) of the user who is currently signed in does not match any custom defined Shell Launcher configurations.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes

WINDOWS EDITION	SUPPORTED
Windows 10 Education	Yes

## Related topics

[WESL\\_UserSetting](#)

[Shell Launcher](#)

# WESL\_UserSetting.GetDefaultShell

10/2/2018 • 2 minutes to read • [Edit Online](#)

This method retrieves the default Shell Launcher configuration.

## Syntax

```
[Static] uint32 GetDefaultShell (
    [Out, Required] string Shell,
    [Out, Required] sint32 DefaultAction
);
```

## Parameters

*Shell* [out, required] The application or executable that Shell Launcher starts as the shell.

*DefaultAction* [out, required] The default action Shell Launcher takes when the shell application exits.

The possible actions are defined in the following table:

VALUE	DESCRIPTION
0	Restarts the shell application.
1	Restarts the device.
2	Shuts down the device.

## Return Value

Returns an HRESULT value that indicates [WMI status](#) or a [WMI error](#).

## Remarks

Shell Launcher uses the default configuration when the security identifier (SID) of the user who is currently signed in does not match any custom defined Shell Launcher configurations.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes

WINDOWS EDITION	SUPPORTED
Windows 10 Education	Yes

## Related topics

[WESL\\_UserSetting](#)

[Shell Launcher](#)

# WESL\_UserSetting.IsEnabled

10/2/2018 • 2 minutes to read • [Edit Online](#)

This method retrieves a value that indicates if Shell Launcher is enabled or disabled.

## Syntax

```
[Static] uint32 IsEnabled(  
    [Out, Required] boolean Enabled  
>);
```

## Parameters

*Enabled* [out, required] A Boolean value that indicates if Shell Launcher is enabled.

## Return Value

Returns an HRESULT value that indicates [WMI status](#) or a [WMI error](#).

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[WESL\\_UserSetting](#)

[Shell Launcher](#)

# WESL\_UserSetting.SetEnabled

10/2/2018 • 2 minutes to read • [Edit Online](#)

This method enables or disables Shell Launcher.

## Syntax

```
[Static] uint32 SetEnabled(  
    [In, Required] boolean Enabled  
>);
```

## Parameters

### *Enabled*

[in, required] A Boolean value that indicates whether to enable or disable Shell Launcher.

## Return Value

Returns an HRESULT value that indicates [WMI status](#) or a [WMI error](#).

## Remarks

This method enables or disables Shell Launcher by modifying the **Shell** value in the registry key **HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon**. If Unified Write Filter (UWF) is enabled, you may need to disable UWF or commit this registry key by using [UWF\\_RegistryFilter.CommitRegistry](#) in order to enable or disable Shell Launcher.

Enabling or disabling Shell Launcher does not take effect until a user signs in.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[WESL\\_UserSetting](#)

[Shell Launcher](#)

# Unbranded Boot

10/2/2018 • 5 minutes to read • [Edit Online](#)

You can suppress Windows elements that appear when Windows starts or resumes and can suppress the crash screen when Windows encounters an error that it cannot recover from. This feature is known as Unbranded Boot.

## IMPORTANT

The first user to sign in to the device must be an administrator. This ensures that the **RunOnce** registry settings correctly apply the settings. Also, when using auto sign-in, you must not configure auto sign-in on your device at design time. Instead, auto sign-in should be configured manually after first signing in as an administrator.

## Requirements

Windows 10 Enterprise, Windows 10 Professional, or Windows 10 Education.

## Terminology

- **Turn on, Enable:** To make the setting available to the device and optionally apply the settings to the device. Generally "turn on" is used in the user interface or control panel, whereas "enable" is used for command line.
- **Configure:** To customize the setting or sub-settings.
- **Embedded Boot Experience:** this feature is called "Embedded Boot Experience" in Windows 10, build 1511.
- **Custom Boot Experience:** this feature is called "Custom Boot Experience" in Windows 10, build 1607 and later.

## Turn on Unbranded Boot settings

Unbranded Boot is an optional component and is not enabled by default in Windows 10. It must be enabled prior to configuring. For end-users, Unbranded Boot is available through **Control Panel > Programs > Programs and Features > Turn Windows features on or off**.

If Windows has already been installed you cannot apply a provisioning package to configure Unbranded Boot; instead you must use BCDEdit to configure Unbranded boot if Windows is installed.

BCDEdit is the primary tool for editing the startup configuration and is on your development computer in the %WINDIR%\System32 folder. You have administrator rights for it. BCDEdit is included in a typical Windows Preinstallation Environment (Windows PE) 4.0. You can download it from the [BCDEdit Commands for Boot Environment](#) in the Microsoft Download Center if needed.

### Turn on Unbranded Boot by using Control Panel

1. In the **Search the web and Windows** field, type Programs and Features and either press Enter or tap or click **Programs and Features** to open it.
2. In the **Programs and Features** window, click **Turn Windows features on or off**.
3. In the **Windows Features** window, expand the **Device Lockdown** node, and check or clear the checkbox for **Unbranded Boot**.
4. Click **OK**. The **Windows Features** window indicates Windows is searching for required files and displays a

progress bar. Once found, the window indicates Windows is applying the changes. When completed, the window indicates the requested changes are completed.

5. Click **Close** to close the **Windows Features** window.

## Configure Unbranded Boot settings at runtime using BCDEdit

1. Open a command prompt as an administrator.
2. To disable the F8 key during startup to prevent access to the **Advanced startup options** menu, type the following:

```
bcdeedit.exe -set {globalsettings} advancedoptions false
```

3. To disable the F10 key during startup to prevent access to the **Advanced startup options** menu, type the following:

```
bcdeedit.exe -set {globalsettings} optionsedit false
```

4. To suppress all Windows UI elements (logo, status indicator, and status message) during startup, type the following:

```
bcdeedit.exe -set {globalsettings} bootuxdisabled on
```

## Configure Unbranded Boot using Unattend

You can also configure the Unattend settings in the [Microsoft-Windows-Embedded-BootExp](#) component to add Unbranded Boot features to your image during the design or imaging phase. You can manually create an Unattend answer file or use Windows System Image Manager (Windows SIM) to add the appropriate settings to your answer file. For more information about the Unbranded Boot settings and XML examples, see the settings in Microsoft-Windows-Embedded-BootExp.

### Unbranded Boot settings

The following table shows Unbranded Boot settings and their values.

SETTING	DESCRIPTION
DisableBootMenu	Contains an integer that disables the F8 and F10 keys during startup to prevent access to the Advanced startup options menu. Set to 1 to disable the menu; otherwise; set to 0 (zero). The default value is 0.
DisplayDisabled	Contains an integer that configures the device to display a blank screen when Windows encounters an error that it cannot recover from. Set to 1 to display a blank screen on error; otherwise; set to 0 (zero). The default value is 0.
HideAllBootUI	Contains an integer that suppresses all Windows UI elements (logo, status indicator, and status message) during startup. Set to 1 to suppress all Windows UI elements during startup; otherwise; set to 0 (zero). The default value is 0.

SETTING	DESCRIPTION
HideBootLogo	Contains an integer that suppresses the default Windows logo that displays during the OS loading phase. Set to 1 to suppress the default Windows logo; otherwise; set to 0 (zero). The default value is 0.
HideBootStatusIndicator	Contains an integer that suppresses the status indicator that displays during the OS loading phase. Set to 1 to suppress the status indicator; otherwise; set to 0 (zero). The default value is 0.
HideBootStatusMessage	Contains an integer that suppresses the startup status text that displays during the OS loading phase. Set to 1 to suppress the startup status text; otherwise; set to 0 (zero). The default value is 0.

## Customize the boot screen using Windows Configuration Designer and Deployment Image Servicing and Management (DISM)

If Windows has not been installed and you are using Windows Configuration Designer to create installation media with settings for Unbranded Boot included in the image, or you are applying a provisioning package during setup, you must enable Unbranded Boot on the installation media with DISM in order for a provisioning package to successfully apply. First you have to create the image or package.

1. Create a provisioning package or create a new Windows image in Windows Configuration Designer by following the instructions in [Create a provisioning package](#).
2. In the Available customizations page, select **Runtime settings** > **SMISettings** and then set the value for the boot screen settings. The following values are just examples.

- **HideAllBootUI**=FALSE
- **HideBootLogo**=FALSE
- **HideBootStatusIndicator**=TRUE
- **HideBootStatusMessage**=TRUE
- **CrashDumpEnabled**=Full dump

### TIP

See [SMISettings](#) in the Windows Configuration Designer reference for more information about the available SMISettings.

3. Once you have finished configuring the settings and building the package or image, you use DISM to apply the settings.
  - a. Open a command prompt with administrator privileges.
  - b. Copy install.wim to a temporary folder on hard drive (in the following steps, it assumes it's called c:\wim).
  - c. Create a new directory.

```
md c:\wim
```

d. Mount the image.

```
dism /mount-wim /wimfile:c:\bootmedia\sources\install.wim /index:1 /MountDir:c:\wim
```

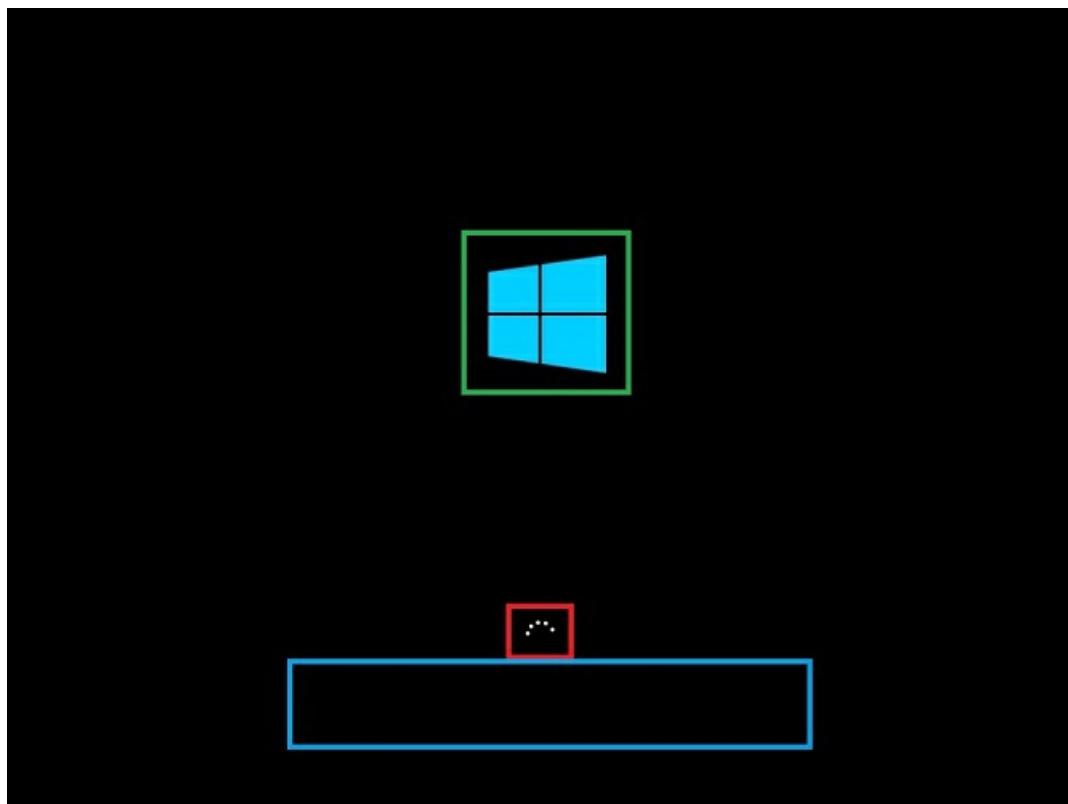
e. Enable the feature.

```
dism /image:c:\wim /enable-feature /featureName:Client-EmbeddedBootExp
```

f. Commit the change.

```
dism /unmount-wim /MountDir:c:\wim /Commit
```

In the following image, the BootLogo is identified by the green outline, the BootStatusIndicator is identified by the red outline, and the BootStatusMessage is identified by the blue outline.



## Replace the startup logo

The only supported way to replace the startup logo with a custom logo is to modify the Boot Graphics Resource Table (BGRT) on a device that uses UEFI as the firmware interface. If your device uses the BGRT to include a custom logo, it is always displayed and you cannot suppress the custom logo.

## Related topics

[Custom Logon](#)

# Unified Write Filter (UWF) feature

10/8/2018 • 4 minutes to read • [Edit Online](#)

Unified Write Filter (UWF) is an optional Windows 10 feature that helps to protect your drives by intercepting and redirecting any writes to the drive (app installations, settings changes, saved data) to a virtual overlay. The virtual overlay is a temporary location that is usually cleared during a reboot or when a guest user logs off.

## Benefits

- Provides a clean experience for thin clients and workspaces that have frequent guests, like school, library or hotel computers. Guests can work, change settings, and install software. After the device reboots, the next guest receives a clean experience.
- Increases security and reliability for kiosks, IoT-embedded devices, or other devices where new apps are not expected to be frequently added.
- Can be used to reduce wear on solid-state drives and other write-sensitive media.

UWF replaces the Windows 7 Enhanced Write Filter (EWF) and the File Based Write Filter (FBWF).

## Features

- UWF can protect most supported writable storage types, including physical hard disks, solid-state drives, internal USB devices, and external SATA devices. You cannot use UWF to protect external removable drives, USB devices or flash drives. Supports both master boot record (MBR) and GUID partition table (GPT) volumes.
- You can use UWF to make read-only media appear to the OS as a writable volume.
- You can manage UWF directly on a Windows 10 device using [uwfmgr.exe](#), or remotely using MDM tools like Microsoft Intune using the [UnifiedWriteFilter CSP](#) or the [UWF WMI](#).
- You can [update and service UWF-protected devices](#), either by using UWF servicing mode or by adding file and registry exclusions to specific system areas.
- On Windows 10, version 1803, you can use a [persistent overlay](#) to allow data saved in the virtual overlay to remain even after a reboot.
- On devices with a disk overlay, you can use [freespace passthrough](#) to access your drive's additional free space.
- UWF supports paging to increase virtual memory, if the page file exists on an unprotected volume. When paging is used together with a RAM-based overlay, the uptime of the system can be significantly increased.

## Requirements

Windows 10 Enterprise, Windows 10 Education, or Windows 10 IoT Core Enterprise

## Limitations

- File systems:

- FAT: fully supported.
- NTFS: fully supported. However, during device startup, NTFS file system journal files can write to a protected volume before UWF has started protecting the volume.
- Other file systems (example: exFAT): You can protect the volume, but cannot create file exclusions or do file commit operations on the volume. Writes to excluded files still influence the growth of the Overlay.
- The overlay does not mirror the entire volume, but dynamically grows to keep track of redirected writes.
- UWF supports up to 16 terabytes of protected volumes.
- UWF does not support the use of fast startup when shutting down your device. If fast startup is turned on, shutting down the device does not clear the overlay. You can disable fast startup in Control Panel by navigating to **Control Panel > All Control Panel Items > Power Options > System Settings** and clearing the checkbox next to **Turn on fast startup (recommended)**.
- UWF does not support [Storage Spaces](#).

## Turn on and configure UWF

UWF is an optional component and is not enabled by default in Windows 10. You must [turn on UWF](#) before you can configure it.

## UWF overlay

You can choose where the overlay is stored (RAM or disk), how much space is reserved, whether the overlay persists after a reboot.

To increase uptime, set up monitoring to check if your overlay is filling up. At certain levels, your device can warn users and/or reboot the device.

To learn more, see [UWF Overlay location and size](#).

## Volumes

A volume is a logical unit that represents an area of persistent storage to the file system that is used by the OS. A volume can correspond to a single physical storage device, such as a hard disk, but volumes can also correspond to a single partition on a physical storage device with multiple partitions, or can span across multiple physical storage devices. For example, a collection of hard disks in a RAID array can be represented as a single volume to the OS.

When you configure UWF to protect a volume, you can specify the volume by using either a drive letter or the volume device identifier. To determine the device identifier for a volume, query the **DeviceID** property in the **Win32\_Volume** WMI class.

If you specify a volume using a drive letter, UWF uses *loose binding* to recognize the volume. By using loose binding, drive letters can be assigned to different volumes if the hardware or volume configuration changes. If you specify a volume using the volume device identifier, UWF uses *tight binding* to recognize the volume. By using tight binding, the device identifier is unique to the storage volume and is independent from the drive letter assigned to the volume by the file system.

## Exclusions

If you want to protect a volume with UWF while excluding specific files, folders, or registry keys from being filtered by UWF, you can add them to a [write filter exclusion](#) list.

## UWF servicing mode

When a device is protected with UWF, you must use UWF servicing mode commands to service the device and apply updates to an image. You can use UWF servicing mode to apply Windows updates, antimalware signature file updates, and custom software or third-party software updates.

For more information about how to use UWF servicing mode to apply software updates to your device, see [Service UWF-protected devices](#).

## Troubleshooting UWF

UWF uses Windows Event Log to log events, errors and messages related to overlay consumption, configuration changes, and servicing.

For more information about how to find event log information for troubleshooting problems with Unified Write Filter (UWF), see [Troubleshooting Unified Write Filter \(UWF\)](#).

## Related topics

[Unbranded Boot](#)

[Custom Logon](#)

[Shell Launcher](#)

# Hibernate Once/Resume Many (HORM)

10/2/2018 • 3 minutes to read • [Edit Online](#)

You can use the Hibernate Once/Resume Many (HORM) feature with Unified Write Filter (UWF) to start your device in a preconfigured state. When HORM is enabled, your system always resumes and restarts from the last saved hibernation file (hiberfil.sys).

A device with HORM enabled can quickly be turned off or shut down, and then restarted into the preconfigured state, even in the event of a sudden power loss.

## NOTE

HORM can be used on Unified Extensible Firmware Interface (UEFI) devices running Windows 10, version 1709, or newer versions of Windows, only. In previous Windows versions, the installation procedure for UEFI creates a hidden system partition. Because UWF cannot protect hidden partitions, HORM cannot be used on any devices that contain a hidden partition, including UEFI-capable devices on older versions of Windows.

## Requirements

Windows 10 Enterprise, Windows 10 Education, or Windows IoT Core (IoT Core). Supported on x86-based and x64-based devices.

## UWF configuration

UWF must be enabled before you can enable or disable HORM. UWF must be configured in the following ways to protect the hibernation file from becoming invalid:

- All fixed volumes that are mounted on the system must be protected by UWF.
- Your system must not have any file, folder, or registry exclusions configured for UWF.
- The UWF overlay must be configured to use RAM mode. HORM does not support disk-backed overlays.

UWF does not filter hibernation files from being written to disk. If you want to protect the preconfigured state of your device, lock down any functionality that can modify the hibernation file. For example, disable hibernation, hybrid sleep, and fast startup on your device for standard user accounts so that the saved hibernation file is not overwritten when entering a sleep, hibernate, or shutdown state.

## Configure HORM

1. On the device, open a command prompt as an administrator.
2. To enable hibernation on the device, type the following command:

```
powercfg /h on
```

3. To enable UWF on your device, type the following command:

```
ufwmgr.exe filter enable
```

4. To protect all volumes on your device, type the following command:

```
ufwmgr.exe volume protect all
```

**NOTE**

DVD RW and floppy drives throw an expected error that can be safely ignored.

5. To restart your device to enable UWF, type the following command:

```
ufwmgr.exe filter restart
```

6. After the device restarts, to verify the UWF changes that you made on your device, type the following command:

```
ufwmgr.exe get-config
```

7. To enable HORM on your device, type the following command:

```
ufwmgr.exe filter enable-horm
```

**NOTE**

Remove all file and registry exclusions before you enable HORM.

8. (Optional) In Control Panel, set the Power Option **When I press the power button** to avoid displaying the command prompt when resuming from hibernation, or use a script to close the command prompt on startup.

9. To hibernate the system one time to create an initial hibernation file, at the command prompt, type the following command:

```
shutdown /h
```

10. Press the power button to wake the system from hibernation.

11. After the system starts from hibernation to create an initial hibernation file, to shut down and restart the system, type the following command:

```
ufwmgr.exe restart
```

12. When HORM is enabled, you cannot change the UWF configuration. To make changes, you must first disable HORM. To disable HORM, type the following command:

```
ufwmgr.exe filter disable-horm
```

13. To restart the system to finish disabling HORM, type the following command:

```
ufwmgr.exe restart
```

The system will restart normally with HORM disabled.

**WARNING**

Do not uninstall UWF when the filter is enabled or when HORM is enabled, either online or offline by using Windows PE.

## Fix an issue when you cannot disable HORM

In rare circumstances, your device can enter a state where you cannot disable HORM normally.

If you cannot disable HORM on your device, use following procedure to resolve this issue:

1. Start your device in Windows PE.
2. Type the following command:

```
bcdedit.exe /set {bootmgr} custom:26000024 0
```

3. Restart the device:

```
shutdown /r/t 0
```

4. Disable HORM:

```
uwmgr.exe filter disable-horm
```

5. Enable HORM:

```
uwmgr.exe filter enable-horm
```

6. Hibernate the device:

```
shutdown /h
```

# Write filter exclusions

10/2/2018 • 5 minutes to read • [Edit Online](#)

You can add specific files or folders on a protected volume to a file exclusion list to exclude those files and folders from being filtered by UWF. When a file or folder is in the exclusion list for a volume, all writes to that file or folder bypass UWF filtering, and are written directly to the protected volume and persist after the device restarts.

You must use an administrator account to add or remove file or folder exclusions during run time, and you must restart the device for new exclusions to take effect.

## IMPORTANT

You cannot add exclusions for the following items:

- `\Windows\System32\config\DEFAULT`
- `\Windows\System32\config\SAM`
- `\Windows\System32\config\SECURITY`
- `\Windows\System32\config\SOFTWARE`
- `\Windows\System32\config\SYSTEM`
- `\Users\<User Name>\NTUSER.DAT`

You also cannot add exclusions for the following items:

- The volume root. For example, C: or D:.
- The `\Windows` folder on the system volume.
- The `\Windows\System32` folder on the system volume.
- The `\Windows\System32\Drivers` folder on the system volume.
- Paging files.

However, you can exclude subdirectories and files under these items.

You cannot rename or move a file or folder from a protected location to an unprotected location, or vice versa.

When write filters are active and you attempt to delete an excluded file or folder in Windows Explorer, the system attempts to move the file or folder to the Recycle Bin. This causes an error, because you cannot move files that are not filtered to a location that is write filter protected.

To work around this, you can disable the Recycle Bin. Alternatively, the user can press Ctrl+Shift and then left-click on the file to directly delete the excluded file, bypassing the Recycle Bin, or the user can delete the excluded file directly from a command prompt. You must restart the device for new exclusions to take effect.

## Virtual Hard Disk (VHD) file exclusions

When you deploy a Windows 10 Enterprise image with UWF on a VHD boot disk, you can protect the volume that contains the VHD file by adding a file exclusion for the VHD file before enabling UWF and protecting the volume.

To add a file exclusion for the VHD file at an administrator command prompt:

```
ufwmgr.exe file add-exclusion <drive containing VHD file>:\<path to VHD file>\<VHD file name>.vhf
```

For example:

```
ufmgr.exe file add-exclusion E:\VHD\test.vhd
```

## Registry exclusions

You can add specific registry keys to an exclusion list to exclude those keys from being filtered by UWF. When a registry key is in the exclusion list, all writes to that registry key bypass UWF filtering and are written directly to the registry and persist after the device restarts.

You must use an administrator account to add or remove registry exclusions during run time, and you must restart the device for new exclusions to take effect.

If you exclude a registry key, all its subkeys are also excluded from filtering. You can exclude registry subkeys only under the following registry keys:

- HKEY\LOCAL\MACHINE\BCD00000000
- HKEY\LOCAL\MACHINE\SYSTEM
- HKEY\LOCAL\MACHINE\SOFTWARE
- HKEY\LOCAL\MACHINE\SAM
- HKEY\LOCAL\MACHINE\SECURITY
- HKEY\LOCAL\MACHINE\COMPONENTS

### NOTE

UWF automatically excludes certain registry keys from being filtered. These registry keys are primarily related to UWF configuration settings and cannot be removed from the exclusion list.

For more information about common registry exclusions, see [Common write filter exclusions](#).

## Common write-filter exclusions

Some services and features write information to a device's persistent volume, and expect that information to be present across device restarts. You may need to configure your write filter to allow for specific file and registry exclusions in order for these services and features to work correctly.

This topic lists registry and file exclusions that can help enable some common services and features to work correctly when write filters are enabled.

If you are running any antivirus or security software in addition to UWF, please consult with your antivirus vendor for advice on how to configure their solution in a UWF environment. You may need to add a UWF exclusion for the signature or update folder.

### Customer Experience Improvement Program (CEIP)

When you choose to participate in the CEIP, your computer or device automatically sends information to Microsoft about how you use certain products. Information from your computer or device is combined with other CEIP data to help Microsoft solve problems and to improve the products and features customers use most often.

CEIP data is stored in files that have a .sqm file name extension. To make sure that the CEIP data in the .sqm files is available on a device that has write filters enabled, you can add file and folder exclusions for the .sqm files and folders.

To locate the .sqm files and folders on your device, search for .sqm files by using File Explorer. Alternately, at a command prompt with administrator rights at the root of the drive, type the following command to obtain a list of .sqm files on the device:

```
dir *.sqm /s
```

Add file and folder exclusions as required for any .sqm files located on your device.

Add registry exclusions for the following registry keys:

- **HKEY\_LOCAL\_MACHINE\Software\Policies\Microsoft\SQMClient\Windows\CEIPEnable**
- **HKEY\_LOCAL\_MACHINE\Software\Microsoft\SQMClient\Windows\CEIPEnable**
- **HKEY\_LOCAL\_MACHINE\Software\Microsoft\SQMClient\UploadDisableFlag**

### **Background Intelligent Transfer Service (BITS)**

Background Intelligent Transfer Service (BITS) downloads or uploads files between a client and server and provides progress information related to the transfers.

Add file exclusions for the following folders and files:

- %ALLUSERSPROFILE%\Microsoft\Network\Downloader

Add registry exclusions for the following registry keys:

- **HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion\BITS\StateIndex**

### **Windows Explorer**

When write filters are active and you attempt to delete an excluded file or folder in Windows Explorer, the system attempts to move the file or folder to the Recycle Bin. This causes an error, because you cannot move files that are not filtered to a location that is write filter protected.

To work around this, you can disable the Recycle Bin. Alternatively, the user can press Ctrl+Shift and then left-click on the file to directly delete the excluded file, bypassing the Recycle Bin, or the user can delete the excluded file directly from a command prompt.

### **Networks**

When you use write filters on your device, you can add file and registry exclusions to enable your device to join wired and wireless networks. The following file and registry exclusions may be required on your device.

Client Group Policy Object (GPO) registry keys:

- Wireless:  
**HKEY\_LOCAL\_MACHINE\SOFTWARE\ Policies\Microsoft\Windows\Wireless\GPTWirelessPolicy**
- Wired: **HKEY\_LOCAL\_MACHINE\SOFTWARE\ Policies\Microsoft\Windows\WiredL2\GP\_Policy**

GPO policy files:

- Wireless: **C:\Windows\wlansvc\Policies**
- Wired: **C:\Windows\dot2svc\Policies**

Interface profile registry keys:

- Wireless: **HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\wlansvc**
- Wired: **HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\dot3svc**

Interface policy file:

- Wireless: **C:\ProgramData\Microsoft\wlansvc\Profiles\Interfaces\{<Interface GUID>\}\{<Profile GUID>\}.xml**
- Wired: **C:\ProgramData\Microsoft\dot3svc\Profiles\Interfaces\{<Interface GUID>\}\{<Profile GUID>\}**

Services registry keys:

- Wireless: **HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\services\Wlansvc**
- Wireless: **HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\services\WwanSvc**
- Wired: **HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\services\dot3svc**

### **Daylight saving time (DST)**

You can add the following registry exclusions to persist daylight saving time (DST) settings on your device.

- **HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Time Zones**
- **HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\TimeZoneInformation**

## Related topics

[Unified Write Filter](#)

[Service UWF-protected devices](#)

[Unified Write Filter WMI provider reference](#)

# Unified Write Filter (UWF) overlay location and size

10/8/2018 • 4 minutes to read • [Edit Online](#)

The Unified Write Filter (UWF) protects the contents of a volume by intercepting write attempts to a protected volume and redirects those write attempts to a virtual overlay.

You can choose where the overlay is stored (RAM or disk), how much space is reserved, and what happens when the overlay fills up.

To increase uptime, set up monitoring to check if your overlay is filling up. At certain levels, your device can warn users and/or reboot the device.

## RAM overlay vs. disk overlay

- **RAM overlay (default):** The virtual overlay is stored in RAM, and is cleared after a reboot.
  - By writing to RAM, you can reduce the wear on write-sensitive media like solid-state drives.
  - RAM is often more limited than drive space. As the drive overlay fills up the available RAM, device performance could be reduced, and users will eventually be prompted to reboot the device. If your users are expected to make many large writes to the overlay, consider using a disk overlay instead.
- **Disk overlay:** The virtual overlay is stored in a temporary location on the drive. By default, the overlay is cleared on reboot.
  - You can use [freespace passthrough](#) to use additional free space on the drive beyond the reserved virtual overlay space.
  - On Windows 10, version 1803, you can use [persistent overlay](#) to allow users to save work in the virtual overlay even after a reboot.

## Overlay size

- Default=1024MB. Set with:
  - **CMD:** `uwfmgr overlay set-size`
  - **CSP:** `NextSession/MaximumOverlaySize`
  - **WMI:** `UWF\Overlay.SetMaximumSize`

When planning device rollouts, we recommend optimizing the overlay size to fit your needs.

For RAM overlays, you'll need to budget some RAM for the system. For example, if the OS requires 2 GB of RAM, and your device has 4 GB of RAM, set the maximum size of the overlay to 2048MB (2 GB) or less.

We recommend enabling UWF on a test device, installing the necessary apps, and putting the device through usage simulations. You can use this Powershell script to find out which files are consuming space:

```
$wmiobject = get-wmiobject -Namespace "root\standardcimv2\embedded" -Class UWF_Overlay
$files = $wmiobject.GetOverlayFiles("C:")
$files.OverlayFiles | select-object -Property FileName,FileSize | export-csv -Path D:\output.csv
```

The amount of overlay used will depend on:

- Device usage patterns.
- Apps that can be accessed. (Some apps have high write volumes and will fill up the overlay faster.)

- Time between resets.
- When files are deleted, UWF removes them from the overlay and returns the freed resources to the available pool.

## Warnings and critical events

As the drive overlay fills up the available space, you can warn your users that they're running out of space, and prompt them to reboot the device or to run a script to clear the overlay.

1. Set warning levels and critical levels (optional). When the overlay is filled to this value, UWF writes an Event Tracing for Windows (ETW) message.

- **Warning level:** Default=512MB. Set with:

- **CMD:** `uwmgr overlay set-warningthreshold`
- **CSP:** `NextSession/WarningOverlayThreshold`
- **WMI:** `UWF\_Overlay.SetWarningThreshold`

- **Critical level:** Default=1024MB. Set with:

- **CMD:** `uwmgr overlay set-criticalthreshold`
- **CSP:** `NextSession/CriticalOverlayThreshold`
- **WMI:** `UWF\_Overlay.SetCriticalThreshold`

Note, these settings will take affect after the next reboot.

2. Use Task Scheduler to detect the ETW message and to warn users to wrap up their work on the device so they do not lose their content before the overlay is cleared. You can also provide a link to script to clear the contents of the overlay.

Create tasks that trigger on the event that the **System** log receives an event ID from **uwfvol**:

OVERLAY USAGE	SOURCE	LEVEL	EVENT ID
Warning threshold	uwfvol	Warning	1
Critical threshold	uwfvol	Error	2
Back to normal	uwfvol	Information	3

3. Reboot the device.

## Freespace passthrough (recommended)

On devices with a disk overlay, you can use freespace passthrough to access your drive's additional free space.

You'll still need to reserve some space on the disk for the overlay. This space is used to manage the overlay, and to store overwrites, such as system updates. All other writes are sent to free space on disk. Over time, the reserved overlay will grow slower and slower, because overwrites will just keep replacing one another.

- **CMD:** `uwmgr overlay set-passthrough (on|off)`

## Persistent overlay

### NOTE

This mode is experimental, and we recommend thoroughly testing it before deploying to multiple devices. This option is not used by default.

On devices with a disk overlay, you can choose to keep working using the overlay data, even after a reboot. This can be helpful in situations where your guest users may need to access for longer periods, and may need to power off

the device between uses.

This option gives your IT department more control over when the overlay is wiped.

To turn persistent overlay on or off:

- **CMD:** `ufwmgr overlay set-persistent (on|off)`

To reset the overlay:

- **CMD:** `ufwmgr overlay reset-persistentstate on`

### Overlay exhaustion

If the size of the overlay is close to or equal to the maximum overlay size, any write attempts will fail, returning an error indicating that there is not enough space to complete the operation. If the overlay on your device reaches this state, your device may become unresponsive and sluggish, and you may need to restart your device.

When Windows shuts down, it attempts to write a number of files to the disk. If the overlay is full, these write attempts fail, causing Windows to attempt to rewrite the files repeatedly until UWF can determine that the device is trying to shut down and resolve the issue. Attempting to shut down by using normal methods when the overlay is full or near to full can result in the device taking a long time, in some cases up to an hour or longer, to shut down.

You can often avoid this issue by using UWF to automatically initiate the shut down or restart:

- **Shut down:**

- **CMD:** `ufwmgr shutdown`
- **CSP:** `ShutdownSystem`
- **WMI:** `UWF\Filter.ShutdownSystem`

- **Restart:**

- **CMD:** `ufwmgr restart`
- **CSP:** `RestartSystem`
- **WMI:** `UWF\Filter.RestartSystem`

## Related topics

[Unified Write Filter](#)

# Unified Write Filter (UWF) overlay location and size

10/8/2018 • 4 minutes to read • [Edit Online](#)

The Unified Write Filter (UWF) protects the contents of a volume by intercepting write attempts to a protected volume and redirects those write attempts to a virtual overlay.

You can choose where the overlay is stored (RAM or disk), how much space is reserved, and what happens when the overlay fills up.

To increase uptime, set up monitoring to check if your overlay is filling up. At certain levels, your device can warn users and/or reboot the device.

## RAM overlay vs. disk overlay

- **RAM overlay (default):** The virtual overlay is stored in RAM, and is cleared after a reboot.
  - By writing to RAM, you can reduce the wear on write-sensitive media like solid-state drives.
  - RAM is often more limited than drive space. As the drive overlay fills up the available RAM, device performance could be reduced, and users will eventually be prompted to reboot the device. If your users are expected to make many large writes to the overlay, consider using a disk overlay instead.
- **Disk overlay:** The virtual overlay is stored in a temporary location on the drive. By default, the overlay is cleared on reboot.
  - You can use [freespace passthrough](#) to use additional free space on the drive beyond the reserved virtual overlay space.
  - On Windows 10, version 1803, you can use [persistent overlay](#) to allow users to save work in the virtual overlay even after a reboot.

## Overlay size

- Default=1024MB. Set with:
  - **CMD:** `ufmgr overlay set-size`
  - **CSP:** `NextSession/MaximumOverlaySize`
  - **WMI:** `UWF\Overlay.SetMaximumSize`

When planning device rollouts, we recommend optimizing the overlay size to fit your needs.

For RAM overlays, you'll need to budget some RAM for the system. For example, if the OS requires 2 GB of RAM, and your device has 4 GB of RAM, set the maximum size of the overlay to 2048MB (2 GB) or less.

We recommend enabling UWF on a test device, installing the necessary apps, and putting the device through usage simulations. You can use this Powershell script to find out which files are consuming space:

```
$wmiobject = get-wmiobject -Namespace "root\standardcimv2\embedded" -Class UWF_Overlay
$files = $wmiobject.GetOverlayFiles("c:")
$files.OverlayFiles | select-object -Property FileName,FileSize | export-csv -Path D:\output.csv
```

The amount of overlay used will depend on:

- Device usage patterns.
- Apps that can be accessed. (Some apps have high write volumes and will fill up the overlay faster.)

- Time between resets.
- When files are deleted, UWF removes them from the overlay and returns the freed resources to the available pool.

## Warnings and critical events

As the drive overlay fills up the available space, you can warn your users that they're running out of space, and prompt them to reboot the device or to run a script to clear the overlay.

1. Set warning levels and critical levels (optional). When the overlay is filled to this value, UWF writes an Event Tracing for Windows (ETW) message.

- **Warning level:** Default=512MB. Set with:

- **CMD:** `ufwmgr overlay set-warningthreshold`
- **CSP:** `NextSession/WarningOverlayThreshold`
- **WMI:** `UWF\_Overlay.SetWarningThreshold`

- **Critical level:** Default=1024MB. Set with:

- **CMD:** `ufwmgr overlay set-criticalthreshold`
- **CSP:** `NextSession/CriticalOverlayThreshold`
- **WMI:** `UWF\_Overlay.SetCriticalThreshold`

Note, these settings will take affect after the next reboot.

2. Use Task Scheduler to detect the ETW message and to warn users to wrap up their work on the device so they do not lose their content before the overlay is cleared. You can also provide a link to script to clear the contents of the overlay.

Create tasks that trigger on the event that the **System** log receives an event ID from **uwfvol**:

OVERLAY USAGE	SOURCE	LEVEL	EVENT ID
Warning threshold	uwfvol	Warning	1
Critical threshold	uwfvol	Error	2
Back to normal	uwfvol	Information	3

3. Reboot the device.

## Freespace passthrough (recommended)

On devices with a disk overlay, you can use freespace passthrough to access your drive's additional free space.

You'll still need to reserve some space on the disk for the overlay. This space is used to manage the overlay, and to store overwrites, such as system updates. All other writes are sent to free space on disk. Over time, the reserved overlay will grow slower and slower, because overwrites will just keep replacing one another.

- **CMD:** `ufwmgr overlay set-passthrough (on|off)`

## Persistent overlay

### NOTE

This mode is experimental, and we recommend thoroughly testing it before deploying to multiple devices. This option is not used by default.

On devices with a disk overlay, you can choose to keep working using the overlay data, even after a reboot. This can be helpful in situations where your guest users may need to access for longer periods, and may need to power

off the device between uses.

This option gives your IT department more control over when the overlay is wiped.

To turn persistent overlay on or off:

- **CMD:** `ufwmgr overlay set-persistent (on|off)`

To reset the overlay:

- **CMD:** `ufwmgr overlay reset-persistentstate on`

### Overlay exhaustion

If the size of the overlay is close to or equal to the maximum overlay size, any write attempts will fail, returning an error indicating that there is not enough space to complete the operation. If the overlay on your device reaches this state, your device may become unresponsive and sluggish, and you may need to restart your device.

When Windows shuts down, it attempts to write a number of files to the disk. If the overlay is full, these write attempts fail, causing Windows to attempt to rewrite the files repeatedly until UWF can determine that the device is trying to shut down and resolve the issue. Attempting to shut down by using normal methods when the overlay is full or near to full can result in the device taking a long time, in some cases up to an hour or longer, to shut down.

You can often avoid this issue by using UWF to automatically initiate the shut down or restart:

- **Shut down:**

- **CMD:** `ufwmgr shutdown`
- **CSP:** `ShutdownSystem`
- **WMI:** `UWF\Filter.ShutdownSystem`

- **Restart:**

- **CMD:** `ufwmgr restart`
- **CSP:** `RestartSystem`
- **WMI:** `UWF\Filter.RestartSystem`

## Related topics

[Unified Write Filter](#)

# Service UWF-protected devices

10/8/2018 • 2 minutes to read • [Edit Online](#)

To update your devices, use UWF servicing mode. UWF servicing mode allows you to apply Windows updates, antimalware signature file updates, and custom software or third-party software updates.

Normally, when the Unified Write Filter (UWF) is active, system updates are disabled, as they would use be erased when the overlay is cleared.

When UWF servicing mode is triggered, Windows does the following:

1. Clears the UWF overlay
2. Reboots the devices
3. Triggers a [system maintenance hour](#).
4. Disables the UWF filter.
5. Scans for and applies Windows updates
6. Scans for and applies app updates from the Microsoft store.
7. After servicing is complete, it re-enables the UWF filter and resumes UWF protection.

## NOTE

Servicing mode requires that all user accounts on the system have a password. If there is a user account that does not include a password, UWF servicing will fail.

## In this section

TOPIC	DESCRIPTION
<a href="#">Antimalware support on UWF-protected devices</a>	Describes the procedures to add support for Windows Defender and System Center Endpoint Protection (SCEP/Forefront) antimalware to your UWF-protected devices.
<a href="#">Apply OEM updates to UWF-protected devices</a>	Provides information about how to apply OEM updates to a UWF-protected device.
<a href="#">Apply Windows updates to UWF-protected devices</a>	Describes the procedures to apply Windows updates to your UWF-protected devices.
<a href="#">UWF master servicing script</a>	Provides information about the UWF master servicing script (UwfServicingMasterScript.cmd).
<a href="#">UWF servicing screen saver</a>	Provides information about how to modify the default UWF servicing screen saver.

# Antimalware support on UWF-protected devices

10/2/2018 • 2 minutes to read • [Edit Online](#)

Learn how to enable antimalware support on your USB Filter-enabled Windows 10 Enterprise device.

When using antimalware software on your Unified Write Filter (UWF)-protected device, you must add the required file and registry exclusions that enable the software to apply updates to signature files and persist changes to the device after a system restart.

## Add support for Windows Defender on UWF-protected devices

Add these exclusions to UWF:

1. File exclusions

- C:\Program Files\Windows Defender
- C:\ProgramData\Microsoft\Windows Defender
- C:\Windows\WindowsUpdate.log
- C:\Windows\Temp\MpCmdRun.log

2. Registry exclusions

- HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows Defender

## Add support for System Center Endpoint Protection on UWF-protected devices

Add these exclusions to UWF:

1. File exclusions

- C:\Program Files\Microsoft Security Client
- C:\Windows\Windowsupdate.log
- C:\Windows\Temp\MpCmdRun.log
- C:\ProgramData\Microsoft\Microsoft Antimalware

2. Registry exclusions

- HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft Antimalware

**NOTE**

Windows 10 Enterprise does not include System Center Endpoint Protection. You can purchase licenses and install System Center Endpoint Protection independently.

## Related topics

[Service UWF-protected devices](#)

# Apply Windows updates to UWF-protected devices

10/2/2018 • 2 minutes to read • [Edit Online](#)

When a device is protected with Unified Write Filter (UWF), you must use UWF servicing mode commands to service the device and apply updates to an image.

UWF servicing mode uses the following files to when it applies Windows updates to your device:

- UWFMgr.exe command-line tool
- UwfServicingScr.scr screen saver
- UwfServicingMasterScript.cmd script

## NOTE

The master servicing script can be modified to service third-party applications, service custom OEM applications, or call custom OEM servicing scripts.

UWF servicing supports the following types of Windows updates:

- Critical updates
- Security updates
- Driver updates

## Apply Windows updates to UWF-protected devices

1. To apply Windows updates to your device, at an administrator command prompt, type the following command:

```
ufwmgr.exe servicing enable
```

2. Restart the device. Use either command.

```
ufwmgr.exe filter restart
```

```
shutdown /r /t 0
```

On restart, the device will automatically sign in to the servicing account and servicing will start.

## IMPORTANT

The default servicing account that is automatically created and used for servicing is named **UWF-Servicing**. It is important that you do not have a user account that has that same name on a device before starting UWF servicing.

Once servicing has started, no user interaction is required. The system may restart if it is required by the Windows updates that are installing. If a restart is required, the system will re-enter servicing mode on restart and continue until all updates have been installed.

While servicing is underway, the UwfServicingScr.scr screen saver displays on the device.

**NOTE**

The UwfServicingScr.scr screen saver that is included with Windows 10 Enterprise is a standard Windows screen saver and can be replaced by a custom OEM screen saver if required.

When Windows update servicing is finished, the system will disable UWF servicing and restart the system with UWF-protection enabled and all file and registry exclusions restored to their original pre-servicing state.

**NOTE**

Be aware that during UWF servicing in Windows 10 Enterprise, Windows Update automatically accepts all Microsoft Software License Terms.

**NOTE**

If Windows updates cannot be installed or return an error, servicing will be disabled and the system will restart with UWF-protection re-enabled and all file and registry exclusions restored to their original pre-servicing state.

## Related topics

[Unified Write Filter](#)

[UWF master servicing script](#)

[UWF servicing screen saver](#)

# Apply OEM updates to UWF-protected devices

10/2/2018 • 2 minutes to read • [Edit Online](#)

To apply OEM updates on a Unified Write Filter (UWF)-protected Windows 10 device, you can modify the UPDATE\_SUCCESS block of UWF master servicing script (UwfServicingMasterScript.cmd) to call a custom OEM script that applies any required OEM updates. The OEM script should return control back to the UWF Master Servicing Script when finished.

The UWF Master Servicing Script (UwfServicingMasterScript.cmd) is located in the \Windows\System32 folder.

## UPDATE\_SUCCESS (UwfServicingMasterScript.cmd)

The UPDATE\_SUCCESS block of the UWF master servicing script follows:

```
:UPDATE_SUCCESS
echo UpdateAgent returned success.
REM
REM echo UpdateAgent executing OEM script
REM OEM can call their custom scripts
REM at this point through a "call".
REM
REM The OEM script should hand control
REM back to this script once it's done.
REM
REM Any error recovery for OEM script
REM should be handled outside of this script
REM post a reboot.
REM
ufwmgr servicing disable
echo Restarting system
goto UPDATE_EXIT
```

## Related topics

[Service UWF-protected devices](#)

[UWF master servicing script](#)

[Unified Write Filter](#)

# UWF master servicing script

10/2/2018 • 2 minutes to read • [Edit Online](#)

The UWF master servicing script (UwfServicingMasterScript.cmd) is located in the \Windows\System32 folder.

## **UwfServicingMasterScript.cmd**

The full UWF master servicing script follows:

```
REM servicing of the device with UWF installed. The script will
REM call UWF manager application to update the system with the
REM latest available updates.
REM The script will detect whether the update operation
REM ended successfully or requires a reboot.
REM
REM The script will change the "SERVICING" state of the device
REM only when the update operation results in a "SUCCESS".
REM A state change of the device requires a reboot.
REM
REM If the update operation requires a "REBOOT" the script will
REM reboot device without changing the "SERVICING" state. The
REM Will then run again on the following reboot until
REM the update operation either return a "SUCCESS" or a "ERROR"
REM
REM Any third-party script that needs to run before the state
REM change should run in the UPDATE_SUCCESS block
REM
REM Environment :
REM It is expected that UWF is turned "OFF", "SERVICING" mode
REM enabled and all other preconditions
REM for servicing are in place.
REM
REM
REM
```

```
echo UpdateAgent starting.
ufwmgr servicing update-windows
if ERRORLEVEL 3010 goto UPDATE_REBOOT
if ERRORLEVEL 0 goto UPDATE_SUCCESS
echo UpdateAgent returned error =%ERRORLEVEL%

:UPDATE_ERROR
ufwmgr servicing disable
echo Restarting system
goto UPDATE_EXIT

:UPDATE_REBOOT
echo UpdateAgent requires a reboot.
echo UpdateAgent restarting system
goto UPDATE_EXIT

:UPDATE_SUCCESS
echo UpdateAgent returned success.
REM
REM echo UpdateAgent executing OEM script
REM OEM can call their custom scripts
REM at this point through a "call".
REM
REM The OEM script should hand control
REM back to this script once it is done.
REM
REM Any error recovery for OEM script
REM should be handled outside of this script
REM post a reboot.
REM
ufwmgr servicing disable
echo Restarting system
goto UPDATE_EXIT

:UPDATE_EXIT
echo UpdateAgent exiting.
shutdown -r -t 5
EXIT /B
```

## Related topics

[Service UWF-protected devices](#)

[Unified Write Filter](#)

# UWF servicing screen saver

10/2/2018 • 2 minutes to read • [Edit Online](#)

The default settings for the Unified Write Filter (UWF) servicing screen saver can be changed through the Windows registry to use custom text, title, font, and color settings.

The UWF servicing screen saver (UwfServicingScr.scr) is located in the \Windows\System32 folder.

## IMPORTANT

When UWF is installed on your device, when you right-click on the **Desktop**, and then click **Personalize > Screen Saver**, the UWF servicing screen saver will appear in the list of available screen savers in the **Screen Saver Settings** dialog box.

Do not select **UwfServicingScr** as the screen saver and then click **Preview**, as you will not be able to exit the UWF servicing screen saver by moving the mouse or pressing a key. The only way to exit the UWF servicing screen saver in this case is by pressing the Ctrl+Alt+Delete keys.

## Modify the default registry settings for the UWF servicing screen saver

1. To modify the default registry settings for the UWF servicing screen saver, from the example shown here, change the values in a text editor, and then save as a .reg file (for example, Overridescreensaver.reg).

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows Embedded\ServicingScreenSaver]
"ColorBackground"=dword:000000ff
"ColorText"=dword:0000ff00
"ColorProgress"=dword:00ff0000
"ScreenSaverTitle"="Device"
"ScreenSaverSubTitle"="Servicing device..."
"HideScreenSaverText"=dword:00000000
"HideScreenSaverProgress"=dword:00000000
"Font"="Algerian"
```

2. On the device, open a command prompt as an administrator. For Windows Shell, to open a command prompt, do the following:
  - a. In Windows Explorer, move to \Windows\System32, right-click **cmd.exe**, and then click **Run as Administrator**.
  - b. Accept the UAC prompt.
3. To apply the custom registry settings for the screen saver to the device, type the following command:

```
regedit.exe /s overridescreensaver.reg
```

The next time the device enters UWF servicing mode, the UwfServicingScr.scr screen saver will use the custom settings.

## Related topics

[Service UWF-protected devices](#)

[Unified Write Filter](#)

# Troubleshooting Unified Write Filter (UWF)

10/2/2018 • 2 minutes to read • [Edit Online](#)

Review the log files and error message information locations for Unified Write Filter (UWF) on your Windows 10 Enterprise device.

If you are having difficulties configuring Unified Write Filter (UWF) on your device, see the following information about how to find event log and error message information for troubleshooting problems with UWF.

## Event logs

UWF uses Windows Event Log to log events, errors and messages.

- Events related to overlay consumption are sent by UWF kernel mode components and are logged in the **Windows Logs\System** event log.
- Event related to configuration changes and servicing logs are sent by UWF user mode components:
  - Error messages are logged in the **Applications and Services Logs\Microsoft\Windows\UnifiedWriteFilter\Admin** event log.
  - Informational messages are logged in the **Applications and Services Logs\Microsoft\Windows\UnifiedWriteFilter\Operational** event log.

## Related topics

[Unified Write Filter](#)

[Common write filter exclusions](#)

[Service UWF-protected devices](#)

[Unified Write Filter WMI provider reference](#)

[uwfmgr.exe](#)

# Unified Write Filter WMI provider reference

10/2/2018 • 2 minutes to read • [Edit Online](#)

To help protect physical storage media, you can use the WMI providers for Unified Write Filter (UWF) to configure UWF.

This section describes the WMI provider classes for UWF.

## In this section

- [UWF\\_ExcludedFile](#): A container class that contains the files and folders that are currently in the file exclusion list for a volume protected by UWF.
- [UWF\\_ExcludedRegistryKey](#): A container class that contains the registry keys that are currently in the registry key exclusion list for UWF.
- [UWF\\_Filter](#): Enables or disables Unified Write Filter (UWF), resets configuration settings for UWF, and shuts down or restarts your device.
- [UWF\\_Overlay](#): Contains the current size of the UWF overlay and manages the critical and warning thresholds for the overlay size.
- [UWF\\_OverlayConfig](#): Manages the configuration of the UWF overlay.
- [UWF\\_OverlayFile](#): Displays and configures global settings for the UWF overlay. You can modify the maximum size and the type of the UWF overlay.
- [UWF\\_RegistryFilter](#): Adds or removes registry exclusions from UWF filtering.
- [UWF\\_Servicing](#): Contains properties and methods that enable you to query and control UWF servicing mode.
- [UWF\\_Volume](#): Manages a volume protected by UWF.

### NOTE

We recommend setting the authentication level to PacketIntegrity or PacketPrivacy for remote clients when you connect to WMI providers under root\standardcimv2\embedded when using WMI scripts or applications. For more information about how to use authentication with WMI providers, see this [WMI Enhancements in Windows PowerShell 2.0 CTP](#) on TechNet.

## Requirements

Windows 10 Enterprise, Windows 10 Education, or Windows 10 IoT Core Enterprise

## Related topics

[uwfmgr.exe](#)

# UWF\_ExcludedFile

10/2/2018 • 2 minutes to read • [Edit Online](#)

Contains the files and folders that are currently in the file exclusion list for a volume protected by Unified Write Filter (UWF).

## Syntax

```
class UWF_ExcludedFile {
    [Read] string FileName;
};
```

## Members

The following tables list any methods and properties that belong to this class.

### Properties

PROPERTY	DATA TYPE	QUALIFIER	DESCRIPTION
FileName	string	[read]	The name of the file or folder path in the file exclusion list, including the full path relative to the volume.

### Remarks

UWF\_ExcludedFile does not represent an actual WMI object, and you cannot use this class to get or set file exclusions.

You must use the [UWF\\_Volume.GetExclusions](#) method to retrieve UWF\_ExcludedFile objects.

You can use the [UWF\\_Volume.AddExclusion](#) and [UWF\\_Volume.RemoveExclusion](#) methods to add or remove file and folder exclusions to a volume.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

## Unified Write Filter WMI provider reference

### Unified Write Filter

# UWF\_ExcludedRegistryKey

10/2/2018 • 2 minutes to read • [Edit Online](#)

Contains the registry keys that are currently in the registry key exclusion list for Unified Write Filter (UWF).

## Syntax

```
class UWF_ExcludedRegistryKey {
    [Read] string RegistryKey;
};
```

## Members

The following tables list any methods and properties that belong to this class.

### Properties

PROPERTY	DATA TYPE	QUALIFIER	DESCRIPTION
RegistryKey	string	[read]	The full path of the registry key in the registry key exclusion list.

### Remarks

UWF\_ExcludedRegistryKey does not represent an actual WMI object, and you cannot use this class to get or set registry key exclusions.

You can use the [UWF\\_RegistryFilter.GetExclusions](#) or [UWF\\_RegistryFilter.FindExclusion](#) methods to retrieve UWF\_ExcludedRegistryKey objects.

You can use the [UWF\\_Volume.AddExclusion](#) and [UWF\\_Volume.RemoveExclusion](#) methods to add or remove registry keys to the UWF registry key exclusion list.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[Unified Write Filter WMI provider reference](#)

## Unified Write Filter

# UWF\_Filter

10/2/2018 • 3 minutes to read • [Edit Online](#)

Enables or disables Unified Write Filter (UWF), resets configuration settings for UWF, and shuts down or restarts your device.

## Syntax

```
class UWF_Filter{
    [key] string Id;
    [read] boolean CurrentEnabled;
    [read] boolean NextEnabled;
    UInt32 Enable();
    UInt32 Disable();
    UInt32 ResetSettings();
    UInt32 ShutdownSystem();
    UInt32 RestartSystem();
};
```

## Members

The following tables list any methods and properties that belong to this class.

### Methods

METHODS	DESCRIPTION
<a href="#">UWF_Filter.Enable</a>	Enables UWF on the next restart.
<a href="#">UWF_Filter.Disable</a>	Disables UWF on the next restart.
<a href="#">UWF_Filter.ResetSettings</a>	Restores UWF settings to the original state that was captured at install time.
<a href="#">UWF_Filter.ShutdownSystem</a>	Safely shuts down a system protected by UWF, even if the overlay is full.
<a href="#">UWF_Filter.RestartSystem</a>	Safely restarts a system protected by UWF, even if the overlay is full.

### Properties

PROPERTY	DATA TYPE	QUALIFIERS	DESCRIPTION
<b>Id</b>	string	[key]	A unique ID. This is always set to <b>UWF_Filter</b> .

PROPERTY	DATA TYPE	QUALIFIERS	DESCRIPTION
<b>CurrentEnabled</b>	Boolean	[read]	Indicates if UWF is enabled for the current session.
<b>NextEnabled</b>	Boolean	[read]	Indicates if UWF is enabled after the next restart.

### Remarks

You must use an administrator account to make any changes to the configuration settings for UWF. Users with any kind of account can read the current configuration settings.

## Example

The following example demonstrates how to enable or disable UWF by using the WMI provider in a PowerShell script.

The PowerShell script creates three functions to help enable or disable UWF. It then demonstrates how to use each function.

The first function, `Disable-UWF`, retrieves a WMI object for **UWF\_Filter**, and calls the **Disable()** method to disable UWF after the next device restart.

The second function, `Enable-UWF`, retrieves a WMI object for **UWF\_Filter**, and calls the **Enable()** method to enable UWF after the next device restart.

The third function, `Display-UWFState`, examines the properties of the **UWF\_Filter** object, and prints out the current settings for **UWF\_Filter**.

```
$COMPUTER = "localhost"
$NAMESPACE = "root\standardcimv2\embedded"

# Create a function to disable the Unified Write Filter driver after the next restart.
function Disable-UWF() {

    # Retrieve the UWF_Filter settings.
    $objUWFInstance = Get-WMIObject -namespace $NAMESPACE -class UWF_Filter;

    if(!$objUWFInstance) {
        "Unable to retrieve Unified Write Filter settings."
        return;
    }

    # Call the method to disable UWF after the next restart. This sets the NextEnabled property to false.
    $retval = $objUWFInstance.Disable();

    # Check the return value to verify that the disable is successful
    if ($retval.ReturnValue -eq 0) {
        "Unified Write Filter will be disabled after the next system restart."
    } else {
        "Unknown Error: " + "{0:x0}" -f $retval.ReturnValue
    }
}

# Create a function to enable the Unified Write Filter driver after the next restart.
function Enable-UWF() {
```

```

# Retrieve the UWF_Filter settings.
$objUWFInstance = Get-WMIObject -namespace $NAMESPACE -class UWF_Filter;

if(!$objUWFInstance) {
    "Unable to retrieve Unified Write Filter settings."
    return;
}

# Call the method to enable UWF after the next restart. This sets the NextEnabled property to false.

$retval = $objUWFInstance.Enable();

# Check the return value to verify that the enable is successful
if ($retval.ReturnValue -eq 0) {
    "Unified Write Filter will be enabled after the next system restart."
} else {
    "Unknown Error: " + "{0:x0}" -f $retval.ReturnValue
}
}

# Create a function to display the current settings of the Unified Write Filter driver.
function Display-UWFState() {

# Retrieve the UWF_Filter object
$objUWFInstance = Get-WmiObject -Namespace $NAMESPACE -Class UWF_Filter;

if(!$objUWFInstance) {
    "Unable to retrieve Unified Write Filter settings."
    return;
}

# Check the CurrentEnabled property to see if UWF is enabled in the current session.
if($objUWFInstance.CurrentEnabled) {
    $CurrentStatus = "enabled";
} else {
    $CurrentStatus = "disabled";
}

# Check the NextEnabled property to see if UWF is enabled or disabled after the next system restart.
if($objUWFInstance.NextEnabled) {
    $NextStatus = "enabled";
} else {
    $NextStatus = "disabled";
}
}

# Some examples of how to call the functions

Display-UWFState

"Enabling Unified Write Filter"
Enable-UWF

Display-UWFState

"Disabling Unified Write Filter"
Disable-UWF

Display-UWFState

```

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No

**WINDOWS EDITION****SUPPORTED**

WINDOWS EDITION	SUPPORTED
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[Unified Write Filter WMI provider reference](#)

[Unified Write Filter](#)

# UWF\_Filter.Disable

10/2/2018 • 2 minutes to read • [Edit Online](#)

Disables Unified Write Filter (UWF) on the next restart.

## Syntax

```
UInt32 Disable();
```

## Parameters

None.

## Return Value

Returns an HRESULT value that indicates [WMI status](#) or a [WMI error](#).

## Remarks

You must use an administrator account to disable UWF.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[UWF\\_Filter](#)

[Unified Write Filter](#)

# UWF\_Filter.Enable

10/2/2018 • 2 minutes to read • [Edit Online](#)

Enables Unified Write Filter (UWF) on the next restart.

## Syntax

```
UInt32 Enable();
```

## Parameters

None.

## Return Value

Returns an HRESULT value that indicates [WMI status](#) or a [WMI error](#).

## Remarks

You must use an administrator account to enable UWF.

You must restart your device after you enable or disable UWF before the change takes effect.

The first time you enable UWF on your device, UWF makes the following changes to your system to improve the performance of UWF:

- Paging files are disabled.
- System restore is disabled.
- SuperFetch is disabled.
- File indexing service is turned off.
- Defragmentation service is turned off.
- Fast boot is disabled.
- BCD setting **bootstatuspolicy** is set to **ignoreallfailures**.

You can change these settings after you enable UWF if you want to. For example, you can move the page file location to an unprotected volume and re-enable paging files.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[UWF\\_Filter](#)

[Unified Write Filter](#)

# UWF\_Filter.ResetSettings

10/2/2018 • 2 minutes to read • [Edit Online](#)

Restores UWF settings to the original configuration settings.

## Syntax

```
UInt32 ResetSettings();
```

## Parameters

None.

## Return Value

Returns an HRESULT value that indicates [WMI status](#) or a [WMI error](#).

## Remarks

You must use an administrator account to reset UWF settings.

The original configuration settings are captured the first time that you enable UWF after you add UWF to your device by using [Turn Windows features on or off](#). You can change the original configuration settings by using [Turn Windows features on or off](#) to remove and then add UWF, and then modifying the configuration to the desired state before you enable UWF.

If you added UWF to your device by using SMI settings in an unattend.xml file, the original configuration settings are captured when Windows 10 Enterprise is installed on your device.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[UWF\\_Filter](#)

[Unified Write Filter](#)

# UWF\_Filter.RestartSystem

10/2/2018 • 2 minutes to read • [Edit Online](#)

Safely restarts a system protected by UWF, even if the overlay is full.

## Syntax

```
UInt32 RestartSystem();
```

## Parameters

None.

## Return Value

Returns an HRESULT value that indicates [WMI status](#) or a [WMI error](#).

## Remarks

You must use an administrator account to call this method.

If the overlay is full, or near full, shutting down or restarting the system normally can cause the system to take an extremely long time to shut down. This occurs when the system repeatedly tries to write files during shutdown, which constantly fail due to the overlay being full. You can call this method to safely restart a system by avoiding this scenario.

If the overlay becomes full while the system is performing a large amount of writes, such as copying a large group of files, calling this method can still result in a long shutdown time.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[UWF\\_Filter](#)

[Unified Write Filter](#)

# UWF\_Filter.ShutdownSystem

10/2/2018 • 2 minutes to read • [Edit Online](#)

Safely shuts down a system protected by UWF, even if the overlay is full.

## Syntax

```
UInt32 ShutdownSystem();
```

## Parameters

None.

## Return Value

Returns an HRESULT value that indicates [WMI status](#) or a [WMI error](#).

## Remarks

You must use an administrator account to call this method.

If the overlay is full, or near full, shutting down or restarting the system normally can cause the system to take an extremely long time to shut down. This occurs when the system repeatedly tries to write files during shutdown, which constantly fail due to the overlay being full. You can call this method to safely shutdown a system by avoiding this scenario.

If the overlay becomes full while the system is performing a large amount of writes, such as copying a large group of files, calling this method can still result in a long shutdown time.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[UWF\\_Filter](#)

[Unified Write Filter](#)

# UWF\_Overlay

10/2/2018 • 2 minutes to read • [Edit Online](#)

Contains the current size of the Unified Write Filter (UWF) overlay and manages the critical and warning thresholds for the overlay size.

## Syntax

```
class UWF_Overlay {
    [key] string Id;
    [read] UInt32 OverlayConsumption;
    [read] UInt32 AvailableSpace;
    [read] UInt32 CriticalOverlayThreshold;
    [read] UInt32 WarningOverlayThreshold;

    UInt32 GetOverlayFiles(
        [in] string Volume,
        [out, EmbeddedInstance("UWF_OverlayFile")] string OverlayFiles[]
    );
    UInt32 SetWarningThreshold(
        UInt32 size
    );
    UInt32 SetCriticalThreshold(
        UInt32 size
    );
}
```

## Members

The following tables list any methods and properties that belong to this class.

### Methods

METHODS	DESCRIPTION
<a href="#">UWF_Overlay.GetOverlayFiles</a>	Returns a list of files of a volume that were cached in the UWF overlay.
<a href="#">UWF_Overlay.SetWarningThreshold</a>	Sets the warning threshold for monitoring the size of the UWF overlay.
<a href="#">UWF_Overlay.SetCriticalThreshold</a>	Sets the critical warning threshold for monitoring the size of the UWF overlay.

### Properties

PROPERTY	DATA TYPE	QUALIFIERS	DESCRIPTION
<b>Id</b>	string	[key]	A unique ID. This is always set to <b>UWF_Overlay</b> .

PROPERTY	DATA TYPE	QUALIFIERS	DESCRIPTION
<b>OverlayConsumption</b>	UInt32	[read]	The current size, in megabytes, of the UWF overlay.
<b>AvailableSpace</b>	UInt32	[read]	The amount of free space, in megabytes, available to the UWF overlay.
<b>CriticalOverlayThreshold</b>	UInt32	[read]	The critical threshold size, in megabytes. UWF sends a critical threshold notification event when the UWF overlay size reaches or exceeds this value.
<b>WarningOverlayThreshold</b>	UInt32	[read]	The warning threshold size, in megabytes. UWF sends a warning threshold notification event when the UWF overlay size reaches or exceeds this value.

## Examples

The following example demonstrates how to use the UWF overlay by using the WMI provider in a PowerShell script.

```
$COMPUTER = "localhost"
$NAMESPACE = "root\standardcimv2\embedded"

# Function to set the Unified Write Filter overlay warning threshold

function Set-OverlayWarningThreshold($ThresholdSize) {

    # Retrieve the overlay WMI object

    $OverlayInstance = Get-WMIOBJECT -namespace $NAMESPACE -class UWF_Overlay;

    if (!$OverlayInstance) {
        "Unable to get handle to an instance of the UWF_Overlay class"
        return;
    }

    # Call the instance method to set the warning threshold value

    $retval = $OverlayInstance.SetWarningThreshold($ThresholdSize);

    # Check the return value to verify that setting the warning threshold is successful

    if ($retval.ReturnValue -eq 0) {
        "Overlay warning threshold has been set to " + $ThresholdSize + " MB"
    } else {
        "Unknown Error: " + "{0:x0}" -f $retval.ReturnValue
    }
}
```

```

# Function to set the Unified Write Filter overlay critical threshold

function Set-OverlayCriticalThreshold($ThresholdSize) {

# Retrieve the overlay WMI object

$OverlayInstance = Get-WMIOBJECT -namespace $NAMESPACE -class UWF_Overlay;

if (!$OverlayInstance) {
    "Unable to get handle to an instance of the UWF_Overlay class"
    return;
}

# Call the instance method to set the warning threshold value

$retval = $OverlayInstance.SetCriticalThreshold($ThresholdSize);

# Check the return value to verify that setting the critical threshold is successful

if ($retval.ReturnValue -eq 0) {
    "Overlay critical threshold has been set to " + $ThresholdSize + " MB"
} else {
    "Unknown Error: " + "{0:x0}" -f $retval.ReturnValue
}
}

# Function to print the current overlay information

function Get-OverlayInformation() {

# Retrieve the Overlay WMI object

$OverlayInstance = Get-WMIOBJECT -namespace $NAMESPACE -class UWF_Overlay;

if (!$OverlayInstance) {
    "Unable to get handle to an instance of the UWF_Overlay class"
    return;
}

# Display the current values of the overlay properties

``nOverlay Consumption: " + $OverlayInstance.OverlayConsumption
"Available Space: " + $OverlayInstance.AvailableSpace
"Critical Overlay Threshold: " + $OverlayInstance.CriticalOverlayThreshold
"Warning Overlay Threshold: " + $OverlayInstance.WarningOverlayThreshold
}

# Examples of using these functions

```nSetting the warning threshold to 768 MB."
Set-OverlayWarningThreshold( 768 )

```nSetting the critical threshold to 896 MB."
Set-OverlayCriticalThreshold( 896 )

```nDisplaying the current state of the overlay."
Get-OverlayInformation

```

## Remarks

Only one **UFW\_Overlay** instance exists for a system protected with UWF.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[Unified Write Filter](#)

# UWF\_OverlayConfig

10/8/2018 • 3 minutes to read • [Edit Online](#)

Displays and configures global settings for the Unified Write Filter (UWF) overlay. You can modify the maximum size and the type of the UWF overlay.

## Syntax

```
class UWF_OverlayConfig{
    [key, Read] boolean CurrentSession;
    [read] UInt32 Type;
    [read] SInt32 MaximumSize;

    UInt32 SetType(
        UInt32 type
    );
    UInt32 SetMaximumSize(
        UInt32 size
    );
}
```

## Members

The following tables list the methods and properties that belong to this class.

### Methods

METHOD	DESCRIPTION
<a href="#">UWF_OverlayConfig.SetMaximumSize</a>	Sets the maximum cache size, in megabytes, of the overlay.
<a href="#">UWF_OverlayConfig.SetType</a>	Sets the type of the UWF overlay to either RAM-based or disk-based.

### Properties

PROPERTY	DATA TYPE	QUALIFIERS	DESCRIPTION
<b>CurrentSession</b>	Boolean	[key, read]	Indicates which session the object contains settings for.  Set to <b>True</b> for the current session; set to <b>False</b> for the next session that begins after a restart.

PROPERTY	DATA TYPE	QUALIFIERS	DESCRIPTION
<b>Type</b>	UInt32	[read]	Indicates the type of overlay. Set to <b>0</b> for a RAM-based overlay; set to <b>1</b> for a disk-based overlay.
<b>MaximumSize</b>	SInt32	[read]	Indicates the maximum cache size, in megabytes, of the overlay.

## Remarks

Changes to the overlay configuration take effect on the next restart in which UWF is enabled.

Before you can change the **Type** or **MaximumSize** properties, UWF must be disabled in the current session.

## Example

The following example demonstrates how to change the maximum size or the storage type of the overlay in UWF by using the Windows Management Instrumentation (WMI) provider in a PowerShell script.

The PowerShell script creates two functions to modify the overlay configuration. It then demonstrates how to use the functions. The first function, **Set-OverlaySize**, sets the maximum size of the overlay. The second function, **Set-OverlayType**, sets the type of the overlay to RAM-based or disk-based.

```
$COMPUTER = "localhost"
$NAMESPACE = "root\standardcimv2\embedded"

# Define common parameters

$CommonParams = @{"namespace"=$NAMESPACE; "computer"=$COMPUTER}

function Set-OverlaySize([UInt32] $size) {

    # This function sets the size of the overlay to which file and registry changes are redirected
    # Changes take effect after the next restart

    # $size is the maximum size in MB of the overlay

    # Make sure that UWF is currently disabled

    $UWFFilter = Get-WmiObject -class UWF_Filter @commonParams

    if ($UWFFilter.CurrentEnabled -eq $false) {

        # Get the configuration for the next session after a restart

        $nextConfig = Get-WMIObject -class UWF_OverlayConfig -Filter "CurrentSession = false" @CommonParams;

        if ($nextConfig) {

            # Set the maximum size of the overlay

            $nextConfig.SetMaximumSize($size);
                write-host "Set overlay max size to $size MB."
            }

        } else {
            write-host "UWF must be disabled in the current session before you can change the overlay size."
        }
    }
}
```

```

        }

function Set-OverlayType([UInt32] $overlayType) {
    # This function sets the type of the overlay to which file and registry changes are redirected
    # Changes take effect after the next restart

    # $overlayType is the type of storage that UWF uses to maintain the overlay. 0 = RAM-based; 1 = disk-based.

    $overlayTypeText = @("RAM-based", "disk-based")

    # Make sure that the overlay type is a valid value

    if ($overlayType -eq 0 -or $overlayType -eq 1) {

        # Make sure that UWF is currently disabled

        $UWFFilter = Get-WmiObject -class UWF_Filter @commonParams

        if ($UWFFilter.CurrentEnabled -eq $false) {

            # Get the configuration for the next session after a restart

            $nextConfig = Get-WMIObject -class UWF_OverlayConfig -Filter "CurrentSession = false"
            @CommonParams;

            if ($nextConfig) {

                # Set the type of the overlay

                $nextConfig.GetType($overlayType);
                write-host "Set overlay type to $overlayTypeText[$overlayType]."
            }
        } else {
            write-host "UWF must be disabled in the current session before you can change the overlay type."
        }
    } else {
        write-host "Invalid value for overlay type. Valid values are 0 (RAM-based) or 1 (disk-based)."
    }
}

# The following sample commands demonstrate how to use the functions to change the overlay configuration

$RAMMode = 0
$DiskMode = 1

Set-OverlaySize 2048

Set-OverlayType $DiskMode

```

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[Unified Write Filter WMI provider reference](#)

[Unified Write Filter](#)

# UWF\_OverlayConfig.SetMaximumSize

10/2/2018 • 2 minutes to read • [Edit Online](#)

Sets the maximum cache size of the Unified Write Filter (UWF) overlay.

## Syntax

```
UInt32 SetMaximumSize(  
    UInt32 size  
)
```

## Parameters

*size* An integer that represents the maximum cache size, in megabytes, of the overlay.

## Return Value

Returns an HRESULT value that indicates [WMI status](#) or a [WMI error](#).

## Remarks

When the size of the overlay reaches the *size* value, UWF returns an error for any attempt to write to a protected volume.

If the overlay type is disk-based, your device must meet the following requirements to change the maximum size of the overlay.

- UWF must be disabled in the current session.
- The *size* value must be at least 1024.
- The system volume on your device must have available free space greater than the new maximum size value.

If the overlay type is RAM-based, your device must meet the following requirement to change the maximum size of the overlay.

- UWF must be disabled in the current session.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

`UWF_OverlayConfig`

Unified Write Filter

# UWF\_OverlayConfig.SetType

10/2/2018 • 2 minutes to read • [Edit Online](#)

Sets the type of the Unified Write Filter (UWF) overlay to either RAM-based or disk-based.

## Syntax

```
UInt32 SetType(  
    UInt32 type  
>);
```

## Parameters

*type* The type of overlay. Set to **0** for a RAM-based overlay; set to **1** for a disk-based overlay.

## Return Value

Returns an HRESULT value that indicates [WMI status](#) or a [WMI error](#).

## Remarks

Changes to the overlay type take effect during the next device restart in which UWF is enabled.

When you change the overlay type from RAM-based to disk-based, UWF creates a file on the system volume. The file has a size equal to the **MaximumSize** property of [UWF\\_OverlayConfig](#).

Before you can change the overlay type to disk-based, your device must meet the following requirements.

- UWF must be disabled in the current session.
- The system volume on your device must have available free space greater than the maximum size of the overlay.
- The maximum size of the overlay must be at least 1024 MB.

Before you can change the overlay type to RAM-based, your device must meet the following requirements.

- UWF must be disabled in the current session.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[UWF\\_OverlayConfig](#)

[Overlay for Unified Write Filter \(UWF\)](#)

[Unified Write Filter](#)

# UWF\_OverlayFile

10/2/2018 • 2 minutes to read • [Edit Online](#)

Contains a file that is currently in the overlay for a volume protected by Unified Write Filter (UWF).

## Syntax

```
class UWF_OverlayFile {  
    [read] string FileName;  
    [read] UInt64 FileSize;  
};
```

## Members

The following table lists any properties that belong to this class.

### Properties

PROPERTY	DATA TYPE	QUALIFIER	DESCRIPTION
<b>FileName</b>	string	[read]	The name of the file in the file overlay.
<b>FileSize</b>	UInt64	[read]	The size of the file in the file overlay.

### Remarks

You cannot use the **UWF\_ OverlayFile** class directly to get overlay files. You must use the **UWF\_Overlay.GetOverlayFiles** method to retrieve **UWF\_ OverlayFile** objects.

For more information about specific limitations and conditions when using the **GetOverlayFiles** method, see the **Remarks** section in the [UWF\\_Overlay.GetOverlayFiles](#) topic in the UWF WMI provider technical reference.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[Unified Write Filter WMI provider reference](#)

## Unified Write Filter

# UWF\_Overlay.GetOverlayFiles

10/2/2018 • 2 minutes to read • [Edit Online](#)

Returns a list of files of a volume that were cached in the Unified Write Filter (UWF) overlay.

## Syntax

```
UInt32 GetOverlayFiles(  
    [in] string Volume,  
    [out, EmbeddedInstance("UWF_OverlayFile")] string OverlayFiles[]  
)
```

## Parameters

*Volume* A string that specifies the drive letter or volume name.

*OverlayFiles* An array of **UWF\_OverlayFiles** objects embedded as strings.

## Return Value

Returns an HRESULT value that indicates [WMI status](#) or a [WMI error](#).

## Remarks

You must use an administrator account to access this method.

The **GetOverlayFiles** method is intended to be used as a diagnostic tool.

Do not base decisions about what to commit based on this method's output.

You should be aware of the following limitations:

- This method is only supported on the NTFS file system.
- This method requires a significant amount of free system memory to succeed (in a linear relationship to overlay usage). The method call fails when there is insufficient memory available to complete the call.
- This method requires significant time to complete (in an exponential relationship to overlay usage).
- This method may show files that are affected by seemingly unrelated operations to both registry and file exclusions and commits.

You should also be aware of the following items when you use the **GetOverlayFiles** method:

- Files that were committed with the `ufmgr.exe file commit` command are also contained in the overlay files list.
- Excluded files may be contained in the overlay files list.
- Files that are smaller than the cluster size (for example, 4 KB in most cases) will not be listed even if they are cached in overlay.
- Changes and deletions in excluded directories, excluded files, or excluded registry items add to overlay usage.
- File and registry commits add to overlay usage.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[UWF\\_Overlay](#)

[Unified Write Filter](#)

# UWF\_Overlay.SetCriticalThreshold

10/2/2018 • 2 minutes to read • [Edit Online](#)

Sets the critical threshold for monitoring the size of the Unified Write Filter (UWF) overlay.

## Syntax

```
UInt32 SetCriticalThreshold(  
    UInt32 size  
)
```

## Parameters

*size* An integer that represents the size, in megabytes, of the critical threshold level for the overlay. If *size* is 0 (zero), UWF does not raise critical threshold events.

## Return Value

Returns an HRESULT value that indicates [WMI status](#) or a [WMI error](#).

## Remarks

When the size of the overlay reaches or exceeds the *size* threshold value, UWF writes the following notification event to the event log.

MESSAGE ID	EVENT CODE	MESSAGE TEXT
UWF_OVERLAY_REACHED_CRITICAL_LEVEL	0x80010002L	The UWF overlay size has reached CRITICAL level.

The critical threshold must be higher than the warning threshold.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[UWF\\_Overlay](#)

## Unified Write Filter

# UWF\_Overlay.SetWarningThreshold

10/2/2018 • 2 minutes to read • [Edit Online](#)

Sets the warning threshold for monitoring the size of the Unified Write Filter (UWF) overlay.

## Syntax

```
UInt32 SetWarningThreshold(  
    UInt32 size  
>);
```

## Parameters

*size* An integer that represents the size, in megabytes, of the warning threshold level for the overlay. If *size* is set to 0 (zero), UWF does not raise warning threshold events.

## Return Value

Returns an HRESULT value that indicates [WMI status](#) or a [WMI error](#).

## Remarks

When the size of the overlay reaches or exceeds the *size* threshold value, UWF writes the following notification event to the event log.

MESSAGE ID	EVENT CODE	MESSAGE TEXT
UWF_OVERLAY_REACHED_WARNING_LEVEL	0x80010001L	The UWF overlay size has reached WARNING level.

The warning threshold must be lower than the critical threshold.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[UWF\\_Overlay](#)

Unified Write Filter

# UWF\_RegistryFilter

10/2/2018 • 5 minutes to read • [Edit Online](#)

Adds or removes registry exclusions from Unified Write Filter (UWF) filtering, and also commits registry changes.

## Syntax

```
class UWF_RegistryFilter{
    [key, Read] boolean CurrentSession;
    [Read, Write] boolean PersistDomainSecretKey;
    [Read, Write] boolean PersistTSCAL;

    UInt32 AddExclusion(
        string RegistryKey
    );
    UInt32 RemoveExclusion(
        string RegistryKey
    );
    UInt32 FindExclusion(
        [in] string RegistryKey,
        [out] boolean bFound
    );
    UInt32 GetExclusions(
        [out, EmbeddedInstance("UWF_ExcludedRegistryKey")] string ExcludedKeys[]
    );
    UInt32 CommitRegistry(
        [in] string RegistryKey,
        [in] string ValueName
    );
    UInt32 CommitRegistryDeletion(
        string Registrykey,
        string ValueName
    );
}
```

## Members

The following tables list the methods and properties that belong to this class.

### Methods

METHOD	DESCRIPTION
<a href="#">UWF_RegistryFilter.AddExclusion</a>	Adds a registry key to the registry exclusion list for UWF.
<a href="#">UWF_RegistryFilter.CommitRegistry</a>	Commits changes to the specified registry key and value.
<a href="#">UWF_RegistryFilter.CommitRegistryDeletion</a>	Deletes the specified registry key or registry value and commits the deletion.
<a href="#">UWF_RegistryFilter.FindExclusion</a>	Determines whether a specific registry key is excluded from being filtered by UWF.

METHOD	DESCRIPTION
<a href="#">UWF_RegistryFilter.GetExclusions</a>	Retrieves all registry key exclusions from a system that is protected by UWF.
<a href="#">UWF_RegistryFilter.RemoveExclusion</a>	Removes a registry key from the registry exclusion list for Unified Write Filter (UWF).

## Properties

PROPERTY	DATA TYPE	QUALIFIERS	DESCRIPTION
<b>CurrentSession</b>	Boolean	[key, read]	Indicates which session the object contains settings for.  <b>True</b> if settings are for the current session; <b>False</b> if settings are for the next session that follows a restart.
<b>PersistDomainSecretKey</b>	Boolean	[read, write]	Indicates if the domain secret registry key is in the registry exclusion list. If the registry key is not in the exclusion list, changes are not persisted after a restart.  Set to <b>True</b> to include in the exclusion list; otherwise, set to <b>False</b> .
<b>PersistTSCAL</b>	Boolean	[read, write]	Indicates if the Terminal Server Client Access License (TSCAL) registry key is in the UWF registry exclusion list. If the registry key is not in the exclusion list, changes are not persisted after a restart.  Set to <b>True</b> to include in the exclusion list; otherwise, set to <b>False</b> .

## Remarks

Additions or removals of registry exclusions, including changes to the values of **PersistDomainSecretKey** and **PersistTSCAL**, take effect after the next restart in which UWF is enabled.

You can only add registry keys in the HKLM registry root to the UWF registry exclusion list.

You can also use **UWF\_RegistryFilter** to exclude the domain secret registry key and the TSCAL registry key from UWF filtering.

## Example

The following example demonstrates how to manage UWF registry exclusions by using the Windows Management Instrumentation (WMI) provider in a PowerShell script.

The PowerShell script creates four functions, and then demonstrates how to use them.

The first function, **Get-RegistryExclusions**, displays a list of UWF registry exclusions for both the current session and the next session that follows a restart.

The second function, **Add-RegistryExclusion**, adds a registry entry to the UWF registry exclusion list after you restart the device.

The third function, **Remove-RegistryExclusion**, removes a registry entry from the UWF exclusion list after you restart the device.

The fourth function, **Clear-RegistryExclusions**, removes all UWF registry exclusions. You must restart the device before UWF stops filtering the exclusions.

```
$COMPUTER = "EMBEDDEDDEVICE"
$NAMESPACE = "root\standardcimv2\embedded"

# Define common parameters

$CommonParams = @{"namespace"=$NAMESPACE; "computer"=$COMPUTER}

function Get-RegistryExclusions() {

    # This function lists the UWF registry exclusions, both
    # for the current session as well as the next session after a restart.

    # Get the UWF_RegistryFilter configuration for the current session

    $currentConfig = Get-WMIObject -class UWF_RegistryFilter @CommonParams |
        where {
            $_.CurrentSession -eq $true
        };

    # Get the UWF_RegistryFilter configuration for the next session after a restart

    $nextConfig = Get-WMIObject -class UWF_RegistryFilter @CommonParams |
        where {
            $_.CurrentSession -eq $false
        };

    # Display registry exclusions for the current session

    if ($currentConfig) {

        Write-Host ""
        Write-Host "The following registry entries are currently excluded from UWF filtering:";

        $currentExcludedList = $currentConfig.GetExclusions()

        if ($currentExcludedList.ExcludedKeys) {
            foreach ($registryExclusion in $currentExcludedList.ExcludedKeys) {
                Write-Host "    $registryExclusion.RegistryKey"
            }
        } else {
            Write-Host "    None"
        }
    } else {
        Write-Error "Could not retrieve UWF_RegistryFilter.";
    }
}
```

```

# Display registry exclusions for the next session after a restart

if ($nextConfig) {

    Write-Host ""
    Write-Host "The following registry entries will be excluded from UWF filtering after the next
restart:";

    $nextExcludedList = $nextConfig.GetExclusions()

    if ($nextExcludedList.ExcludedKeys) {
        foreach ($registryExclusion in $nextExcludedList.ExcludedKeys) {
            Write-Host " " $registryExclusion.RegistryKey
        }
    } else {
        Write-Host " None"
    }
    Write-Host ""
}

function Add-RegistryExclusion($exclusion) {

# This function adds a new UWF registry exclusion.
# The new registry exclusion takes effect the next time the device is restarted.

# $exclusion is the path of the registry exclusion

# Get the UWF_RegistryFilter configuration for the next session after a restart

$nextConfig = Get-WMIObject -class UWF_RegistryFilter @CommonParams |
    where {
        $_.CurrentSession -eq $false
    };

# Add the exclusion

if ($nextConfig) {
    $nextConfig.AddExclusion($exclusion) | Out-Null;
    Write-Host "Added exclusion $exclusion.";
} else {
    Write-Error "Could not retrieve UWF_RegistryFilter";
}
}

function Remove-RegistryExclusion($exclusion) {

# This function removes a UWF registry exclusion.
# The registry exclusion is removed the next time the device is restarted

# $exclusion is the path of the registry exclusion

# Get the UWF_RegistryFilter configuration for the next session after a restart

$nextConfig = Get-WMIObject -class UWF_RegistryFilter @CommonParams |
    where {
        $_.CurrentSession -eq $false
    };

# Try to remove the exclusion

if ($nextConfig) {
    try {
        $nextConfig.RemoveExclusion($exclusion) | Out-Null;
        Write-Host "Removed exclusion $exclusion.";
    } catch {
        Write-Host "Could not remove exclusion $exclusion."
    }
} else {
}
}

```

```

        Write-Error "Could not retrieve UWF_RegistryFilter";
    }

function Clear-RegistryExclusions() {

# This function removes all UWF registry exclusions
# The registry exclusions are removed the next time the device is restarted

# Get the configuration for the next session

$nextConfig = Get-WMIObject -class UWF_RegistryFilter @CommonParams |
    where {
        $_.CurrentSession -eq $false
    };

# Remove all registry exclusions

if ($nextConfig) {

    Write-Host "Removing all registry exclusions:";

    $nextExcludedList = $nextConfig.GetExclusions()

    if ($nextExcludedList) {
        foreach ($registryExclusion in $nextExcludedList.ExcludedKeys) {
            Write-Host "Removing:" $registryExclusion.RegistryKey
            $nextConfig.RemoveExclusion($registryExclusion.RegistryKey) | Out-Null
        }
    } else {
        Write-Host "No registry exclusions to remove."
    }
    Write-Host ""
}

# Some examples of using the functions

Clear-RegistryExclusions

Get-RegistryExclusions

Add-RegistryExclusion "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer"
Add-RegistryExclusion "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\DateTime\Servers\
(Default)"

Get-RegistryExclusions

Remove-RegistryExclusion "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer"

Get-RegistryExclusions

Clear-RegistryExclusions

```

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes

WINDOWS EDITION	SUPPORTED
Windows 10 Education	Yes

## Related topics

[Unified Write Filter](#)

# UWF\_RegistryFilter.AddExclusion

10/2/2018 • 2 minutes to read • [Edit Online](#)

Adds a registry key to the registry exclusion list for Unified Write Filter (UWF).

## IMPORTANT

Only registry subkeys under the following registry keys can be added to the exclusion list.

- HKEY\_LOCAL\_MACHINE\BCD00000000
- HKEY\_LOCAL\_MACHINE\SYSTEM
- HKEY\_LOCAL\_MACHINE\SOFTWARE
- HKEY\_LOCAL\_MACHINE\SAM
- HKEY\_LOCAL\_MACHINE\SECURITY
- HKEY\_LOCAL\_MACHINE\COMPONENTS

## IMPORTANT

Excluding a registry key from filtering also excludes all subkeys from filtering.

## Syntax

```
UInt32 AddExclusion(  
    string RegistryKey  
>);
```

## Parameters

*RegistryKey* A string that contains the full path of the registry key.

## Return Value

Returns an HRESULT value that indicates [WMI status](#) or a [WMI error](#).

## Remarks

You must restart the device before the registry key is excluded from UWF filtering.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes

WINDOWS EDITION	SUPPORTED
Windows 10 Education	Yes

## Related topics

[UWF\\_RegistryFilter](#)

[Unified Write Filter](#)

# UWF\_RegistryFilter.CommitRegistry

10/2/2018 • 2 minutes to read • [Edit Online](#)

Commits changes to the specified registry key and value.

## Syntax

```
UInt32 CommitRegistry(  
    [in] string RegistryKey,  
    [in] string ValueName  
)
```

## Parameters

*RegistryKey* A string that contains the full path of the registry key to be committed.

*ValueName* A string that contains the name of the value to be committed.

## Return Value

Returns an HRESULT value that indicates [WMI status](#) or a [WMI error](#).

## Remarks

This method will commit only the value specified by *ValueName* under *RegistryKey* if *ValueName* is specified.

You must use an administrator account to change any properties or call any methods that change the configuration settings.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[UWF\\_RegistryFilter](#)

[Unified Write Filter](#)

# UWF\_RegistryFilter.CommitRegistryDeletion

10/2/2018 • 2 minutes to read • [Edit Online](#)

Deletes the specified registry key or registry value and commits the deletion.

## Syntax

```
UInt32 CommitRegistryDeletion(  
    string Registrykey,  
    string ValueName  
)
```

## Parameters

*RegistryKey* A string that contains the full path of the registry key that contains the value to be deleted. If *ValueName* is empty, the entire registry key is deleted.

*ValueName* A string that contains the name of the value to be deleted.

## Return Value

Returns an HRESULT value that indicates [WMI status](#) or a [WMI error](#).

## Remarks

If *ValueName* is specified, this method will delete only the value specified by *ValueName* that is contained by *RegistryKey*. If *ValueName* is empty, the entire *RegistryKey* and all its sub keys are deleted.

This method deletes the registry key or registry value from both the overlay and the persistent storage.

You must use an administrator account to change any properties or call any methods that change the configuration settings.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[UWF\\_RegistryFilter](#)

[Unified Write Filter](#)

# UWF\_RegistryFilter.FindExclusion

10/2/2018 • 2 minutes to read • [Edit Online](#)

Checks if a specific registry key is excluded from being filtered by Unified Write Filter (UWF).

## Syntax

```
UInt32 FindExclusion(  
    [in] string RegistryKey,  
    [out] boolean bFound  
>);
```

## Parameters

*RegistryKey* [in] A string that contains the full path of the registry key.

*bFound* [out] Indicates if the *RegistryKey* is in the exclusion list of registry keys.

## Return Value

Returns an HRESULT value that indicates [WMI status](#) or a [WMI error](#).

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[UWF\\_RegistryFilter](#)

[Unified Write Filter](#)

# UWF\_RegistryFilter.GetExclusions

10/2/2018 • 2 minutes to read • [Edit Online](#)

Retrieves all registry key exclusions from a device that is protected by Unified Write Filter (UWF).

## Syntax

```
UInt32 GetExclusions(  
    [out, EmbeddedInstance("UWF_ExcludedRegistryKey")] string ExcludedKeys[]  
)
```

## Parameters

*ExcludedKeys* [out] An array of [UWF\\_ExcludedRegistryKey](#) objects that represent the registry keys excluded from UWF filtering.

## Return Value

Returns an HRESULT value that indicates [WMI status](#) or a [WMI error](#).

## Remarks

If this method does not find any registry keys in the registry key exclusion list, it sets the *ExcludedKeys* parameter to null.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[UWF\\_RegistryFilter](#)

[Unified Write Filter](#)

# UWF\_RegistryFilter.RemoveExclusion

10/2/2018 • 2 minutes to read • [Edit Online](#)

Removes a registry key from the registry exclusion list for Unified Write Filter (UWF).

## Syntax

```
UInt32 RemoveExclusion(  
    string RegistryKey  
>);
```

## Parameters

*RegistryKey* A string that contains the full path of the registry key.

## Return Value

Returns an HRESULT value that indicates [WMI status](#) or a [WMI error](#).

## Remarks

You must restart the device before the registry key is excluded from UWF filtering.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[UWF\\_RegistryFilter](#)

[Unified Write Filter](#)

# UWF\_Servicing

10/2/2018 • 2 minutes to read • [Edit Online](#)

This class contains properties and methods that enable you to query and control Unified Write Filter (UWF) servicing mode.

## Syntax

```
class UWF_Servicing {
    [key, read] boolean CurrentSession;
    [read] boolean ServicingEnabled;

    UInt32 Enable();
    UInt32 Disable();
    UInt32 UpdateWindows(
        [out] UInt32 UpdateStatus
    );
}
```

## Members

The following tables list the methods and properties that belong to this class.

### Methods

METHOD	DESCRIPTION
<a href="#">UWF_Servicing.Disable</a>	Disables Unified Write Filter (UWF) servicing mode. The system leaves servicing mode in the next session that follows a restart.
<a href="#">UWF_Servicing.Enable</a>	Enables Unified Write Filter (UWF) servicing mode. The system enters servicing mode in the next session that follows a restart.
<a href="#">UWF_Servicing.UpdateWindows</a>	Calls Windows Update to download and install critical and security updates for your device running Windows 10 Enterprise.

### Properties

PROPERTY	DATA TYPE	QUALIFIERS	DESCRIPTION
----------	-----------	------------	-------------

PROPERTY	DATA TYPE	QUALIFIERS	DESCRIPTION
<b>CurrentSession</b>	Boolean	[key, read]	Indicates when to enable servicing.  <b>True</b> if servicing is enabled in the current session; <b>False</b> if servicing will be enabled in the session that follows a restart.
<b>ServiceEnabled</b>	Boolean	[read]	Indicates if the system is in servicing mode in the current session, or will be in servicing mode in the next session that follows a restart.  <b>True</b> if servicing is enabled; otherwise, <b>False</b> .

## Remarks

This class only has two instances, one for the current session, and another for the next session that follows a restart.

## Example

The following example shows how to enable and disable UWF servicing mode on a device by using the Windows Management Instrumentation (WMI) provider in a PowerShell script.

```

$COMPUTER = "localhost"
$NAMESPACE = "root\standardcimv2\embedded"

# Define common parameters

$CommonParams = @{"namespace"=$NAMESPACE; "computer"=$COMPUTER}

# Enable UWF servicing

$nextSession = Get-WmiObject -class UWF_Servicing @CommonParams | where {
    $_.CurrentSession -eq $false
}

if ($nextSession) {

    $nextSession.Enable() | Out-Null;
    Write-Host "This device is enabled for servicing mode after the next restart."
}

# Disable UWF servicing

$nextSession = Get-WmiObject -class UWF_Servicing @CommonParams | where {
    $_.CurrentSession -eq $false
}

if ($nextSession) {

    $nextSession.Disable() | Out-Null;
    Write-Host "Servicing mode is now disabled for this device."
}

```

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[Unified Write Filter](#)

# UWF\_Servicing.Disable

10/2/2018 • 2 minutes to read • [Edit Online](#)

Disables Unified Write Filter (UWF) servicing mode.

## Syntax

```
UInt32 Disable();
```

## Parameters

None.

## Return Value

Returns an HRESULT value that indicates [WMI status](#) or a [WMI error](#).

## Remarks

When this method is called, the system will leave servicing mode in the next session after a restart.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[UWF\\_Servicing](#)

[Unified Write Filter](#)

# UWF\_Servicing.Enable

10/2/2018 • 2 minutes to read • [Edit Online](#)

Enables Unified Write Filter (UWF) servicing mode.

## Syntax

```
UInt32 Enable();
```

## Parameters

None.

## Return Value

Returns an HRESULT value that indicates [WMI status](#) or a [WMI error](#).

## Remarks

When this method is called, the system will enter servicing mode in the next session after a restart.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[UWF\\_Servicing](#)

[Unified Write Filter](#)

# UWF\_Servicing.UpdateWindows

10/2/2018 • 2 minutes to read • [Edit Online](#)

Calls Windows Update to download and install critical and security updates for your device running Windows 10 Enterprise.

## Syntax

```
UInt32 UpdateWindows(  
    [out] UInt32 UpdateStatus  
) ;
```

## Parameters

*UpdateStatus* [out] An integer that contains the status of the Windows Update operation, according to the following table:

UPDATESTATUS	DESCRIPTION
0	Success.
3010	Restart required.
Any other value.	Generic error.

## Return Value

Returns an HRESULT value that indicates [WMI status](#) or a [WMI error](#).

## Remarks

This method is meant to be used as part of a servicing script. For more information, see [Service UWF-protected devices](#).

This method does not disable or enable Unified Write Filter (UWF). If you call this method while UWF is enabled, updates may be lost when the device restarts.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes

WINDOWS EDITION	SUPPORTED
Windows 10 Education	Yes

## Related topics

[UWF\\_Servicing](#)

[Unified Write Filter](#)

# UWF\_Volume

10/2/2018 • 7 minutes to read • [Edit Online](#)

This class manages a volume protected by Unified Write Filter (UWF).

## Syntax

```
class UWF_Volume {  
    [key, Read] boolean CurrentSession;  
    [key, Read] string DriveLetter;  
    [key, Read] string VolumeName;  
    [Read, Write] boolean BindByDriveLetter;  
    [Read] boolean CommitPending;  
    [Read, Write] boolean Protected;  
  
    UInt32 CommitFile([in] string FilePath);  
    UInt32 CommitFileDeletion(string FileName);  
    UInt32 Protect();  
    UInt32 Unprotect();  
    UInt32 SetBindByDriveLetter(boolean bBindByVolumeName);  
    UInt32 AddExclusion(string FileName);  
    UInt32 RemoveExclusion(string FileName);  
    UInt32 RemoveAllExclusions();  
    UInt32 FindExclusion([in] string FileName, [out] bFound);  
    UInt32 GetExclusions([out, EmbeddedInstance("UWF_ExcludedFile")] string ExcludedFiles[]);  
};
```

## Members

The following tables list the methods and properties that belong to this class.

### Methods

METHOD	DESCRIPTION
<a href="#">UWF_Volume.AddExclusion</a>	Adds a file or folder to the file exclusion list for a volume protected by UWF.
<a href="#">UWF_Volume.CommitFile</a>	Commits changes from the overlay to the physical volume for a specified file on a volume protected by Unified Write Filter (UWF).
<a href="#">UWF_Volume.CommitFileDeletion</a>	Deletes a protected file from the volume, and commits the deletion to the physical volume.
<a href="#">UWF_Volume.FindExclusion</a>	Determines whether a specific file or folder is in the exclusion list for a volume protected by UWF.

METHOD	DESCRIPTION
<a href="#">UWF_Volume.GetExclusions</a>	Retrieves a list of all file exclusions for a volume protected by UWF.
<a href="#">UWF_Volume.Protect</a>	Protects the volume after the next system restart, if UWF is enabled after the restart.
<a href="#">UWF_Volume.RemoveAllExclusions</a>	Removes all files and folders from the file exclusion list for a volume protected by UWF.
<a href="#">UWF_Volume.RemoveExclusion</a>	Removes a specific file or folder from the file exclusion list for a volume protected by UWF.
<a href="#">UWF_Volume.SetBindByDriveLetter</a>	Sets the <b>BindByDriveLetter</b> property, which indicates whether the UWF volume is bound to the physical volume by drive letter or by volume name.
<a href="#">UWF_Volume.Unprotect</a>	Disables UWF protection of the volume after the next system restart.

## Properties

PROPERTY	DATA TYPE	QUALIFIERS	DESCRIPTION
<b>BindByDriveLetter</b>	Boolean	[read, write]	Indicates the type of binding that the volume uses.  Set to <b>True</b> to bind the volume by <b>DriveLetter</b> (loose binding); set to <b>False</b> to bind the volume by <b>VolumeName</b> (tight binding).
<b>CommitPending</b>	Boolean	[read]	Reserved for Microsoft use.
<b>CurrentSession</b>	Boolean	[key, read]	Indicates which session the object contains settings for.  <b>True</b> if settings are for the current session; <b>False</b> if settings are for the next session that follows a restart.

PROPERTY	DATA TYPE	QUALIFIERS	DESCRIPTION
<b>DriveLetter</b>	string	[key, read]	The drive letter of the volume. If the volume does not have a drive letter, this value is <b>NULL</b> .
<b>Protected</b>	Boolean	[read, write]	If <b>CurrentSession</b> is <b>true</b> , indicates whether the volume is currently protected by UWF.  If <b>CurrentSession</b> is <b>false</b> , indicates whether the volume is protected in the next session after the device restarts.
<b>VolumeName</b>	string	[key, read]	The unique identifier of the volume on the current system. The <b>VolumeName</b> is the same as the <b>DeviceID</b> property of the <a href="#">Win32_Volume</a> class for the volume.

## Remarks

You must use an administrator account to change any properties or call any methods that change the configuration settings.

### Turn UWF protection on or off

The following example demonstrates how to protect or unprotect a volume with UWF by using the Windows Management Instrumentation (WMI) provider in a PowerShell script.

The PowerShell script creates a function, **Set-ProtectVolume**, that turns UWF protection on or off for a volume. The script then demonstrates how to use the function.

```

$COMPUTER = "localhost"
$NAMESPACE = "root\standardcimv2\embedded"

# Define common parameters

$CommonParams = @{"namespace"=$NAMESPACE; "computer"=$COMPUTER}

# Create a function to protect or unprotect a volume based on the drive letter of the volume

function Set-ProtectVolume($driveLetter, [bool] $enabled) {

    # Each volume has two entries in UWF_Volume, one for the current session and one for the next session after a
    # restart
    # You can only change the protection status of a drive for the next session

    $nextConfig = Get-WMIObject -class UWF_Volume @CommonParams |
        where {
            $_.DriveLetter -eq "$driveLetter" -and $_.CurrentSession -eq $false
        };

    # If a volume entry is found for the drive letter, enable or disable protection based on the $enabled
    # parameter

    if ($nextConfig) {

        Write-Host "Setting drive protection on $driveLetter to $enabled"

        if ($Enabled -eq $true) {
            $nextConfig.Protect() | Out-Null;
        } else {
            $nextConfig.Unprotect() | Out-Null;
        }
    }

    # If the drive letter does not match a volume, create a new UWF_volume instance

    else {
        Write-Host "Error: Could not find $driveLetter. Protection is not enabled."
    }
}

# The following sample commands demonstrate how to use the Set-ProtectVolume function
# to protect and unprotect volumes

Set-ProtectVolume "C:" $true
Set-ProtectVolume "D:" $true

Set-ProtectVolume "C:" $false

```

## Manage UWF file and folder exclusions

The following example demonstrates how to manage UWF file and folder exclusions by using the WMI provider in a PowerShell script. The PowerShell script creates four functions, and then demonstrates how to use them.

The first function, **Get-FileExclusions**, displays a list of UWF file exclusions that exist on a volume. Exclusions for both the current session and the next session that follows a restart are displayed.

The second function, **Add-FileExclusion**, adds a file or folder to the UWF exclusion list for a given volume. The exclusion is added for the next session that follows a restart.

The third function, **Remove-FileExclusion**, removes a file or folder from the UWF exclusion list for a given volume. The exclusion is removed for the next session that follows a restart.

The fourth function, **Clear-FileExclusions**, removes all UWF file and folder exclusions from a given volume. The exclusions are removed for the next session that follows a restart.

```

$COMPUTER = "localhost"
$NAMESPACE = "root\standardcimv2\embedded"

# Define common parameters

$CommonParams = @{"namespace"=$NAMESPACE; "computer"=$COMPUTER}

function Get-FileExclusions($driveLetter) {

    # This function lists the UWF file exclusions for a volume, both
    # for the current session as well as the next session after a restart

    # $driveLetter is the drive letter of the volume

    # Get the UWF_Volume configuration for the current session

    $currentConfig = Get-WMIObject -class UWF_Volume @CommonParams |
        where {
            $_.DriveLetter -eq "$driveLetter" -and $_.CurrentSession -eq $true
        };

    # Get the UWF_Volume configuration for the next session after a restart

    $nextConfig = Get-WMIObject -class UWF_Volume @CommonParams |
        where {
            $_.DriveLetter -eq "$driveLetter" -and $_.CurrentSession -eq $false
        };

    # Display file exclusions for the current session

    if ($currentConfig) {

        Write-Host "The following files and folders are currently excluded from UWF filtering for
$driveLetter";

        $currentExcludedList = $currentConfig.GetExclusions()

        if ($currentExcludedList) {
            foreach ($fileExclusion in $currentExcludedList.ExcludedFiles) {
                Write-Host " " $fileExclusion.FileName
            }
        } else {
            Write-Host " None"
        }
    } else {
        Write-Error "Could not find drive $driveLetter";
    }

    # Display file exclusions for the next session after a restart

    if ($nextConfig) {

        Write-Host ""
        Write-Host "The following files and folders will be excluded from UWF filtering for $driveLetter
after the next restart:";

        $nextExcludedList = $nextConfig.GetExclusions()

        if ($nextExcludedList) {
            foreach ($fileExclusion in $nextExcludedList.ExcludedFiles) {
                Write-Host " " $fileExclusion.FileName
            }
        } else {
            Write-Host " None"
        }

        Write-Host ""
    }
}

```

```

}

function Add-FileExclusion($driveLetter, $exclusion) {
    # This function adds a new UWF file exclusion to a volume
    # The new file exclusion takes effect the next time the device is restarted and UWF is enabled

    # $driveLetter is the drive letter of the volume
    # $exclusion is the path and filename of the file or folder exclusion

    # Get the configuration for the next session for the volume

    $nextConfig = Get-WMIObject -class UWF_Volume @CommonParams |
        where {
            $_.DriveLetter -eq "$driveLetter" -and $_.CurrentSession -eq $false
        };

    # Add the exclusion

    if ($nextConfig) {
        $nextConfig.AddExclusion($exclusion) | Out-Null;
        Write-Host "Added exclusion $exclusion for $driveLetter";
    } else {
        Write-Error "Could not find drive $driveLetter";
    }
}

function Remove-FileExclusion($driveLetter, $exclusion) {
    # This function removes a UWF file exclusion from a volume
    # The file exclusion is removed the next time the device is restarted

    # $driveLetter is the drive letter of the volume
    # $exclusion is the path and filename of the file or folder exclusion

    # Get the configuration for the next session for the volume

    $nextConfig = Get-WMIObject -class UWF_Volume @CommonParams |
        where {
            $_.DriveLetter -eq "$driveLetter" -and $_.CurrentSession -eq $false
        };

    # Try to remove the exclusion

    if ($nextConfig) {
        try {
            $nextConfig.RemoveExclusion($exclusion) | Out-Null;
            Write-Host "Removed exclusion $exclusion for $driveLetter";
        } catch {
            Write-Host "Could not remove exclusion $exclusion on drive $driveLetter"
        }
    } else {
        Write-Error "Could not find drive $driveLetter";
    }
}

function Clear-FileExclusions($driveLetter) {
    # This function removes all UWF file exclusions on a volume
    # The file exclusions are removed the next time the device is restarted

    # $driveLetter is the drive letter of the volume

    # Get the configuration for the next session for the volume

    $nextConfig = Get-WMIObject -class UWF_Volume @CommonParams |
        where {
            $_.DriveLetter -eq "$driveLetter" -and $_.CurrentSession -eq $false
        };
}

```

```

        }

# Remove all file and folder exclusions

if ($nextConfig) {
    $nextConfig.RemoveAllExclusions() | Out-Null;
    Write-Host "Cleared all exclusions for $driveLetter";
} else {
    Write-Error "Could not clear exclusions for drive $driveLetter";
}

}

# Some examples of using the functions

Clear-FileExclusions "C:"

Add-FileExclusion "C:" "\Users\Public\Public Documents"
Add-FileExclusion "C:" "\myfolder\myfile.txt"

Get-FileExclusions "C:"

Remove-FileExclusion "C:" "\myfolder\myfile.txt"

Get-FileExclusions "C:"

```

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[Unified Write Filter](#)

# UWF\_Volume.AddExclusion

10/2/2018 • 2 minutes to read • [Edit Online](#)

Adds a file or folder to the file exclusion list for a volume protected by Unified Write Filter (UWF).

## Syntax

```
UInt32 AddExclusion(  
    string FileName  
>);
```

## Parameters

*FileName* A string that contains the full path of the file or folder relative to the volume.

## Return Value

Returns an HRESULT value that indicates [WMI status](#) or a [WMI error](#).

## Remarks

You must use an administrator account to add or remove file or folder exclusions during run time, and you must restart the device for new exclusions to take effect.

### IMPORTANT

You can't add exclusions for the following items:

- The volume root. For example, C: or D:.
- The \Windows folder on the system volume.
- The \Windows\System32 folder on the system volume.
- The \Windows\system32\drivers folder on the system volume.
- Paging files.

However, you can exclude subdirectories and files under these items.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[UWF\\_Volume](#)

[Unified Write Filter](#)

# UWF\_Volume.CommitFile

10/2/2018 • 2 minutes to read • [Edit Online](#)

Commits changes from the overlay to the physical volume for a specified file on a volume protected by Unified Write Filter (UWF).

## Syntax

```
UInt32 CommitFile(  
    [in] string FileName  
>;
```

## Parameters

*FileName* [in] A string that contains the path of the file to commit on the overlay, but does not include the drive letter or volume name. For example, “\users\test.dat”.

## Return Value

Returns an HRESULT value that indicates [WMI status](#) or a [WMI error constant](#).

## Remarks

The *FileName* must contain the name of a file that exists. The **CommitFile** method cannot commit a file that does not exist.

You must use an administrator account to change any properties or call any methods that change the configuration settings.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[UWF\\_Volume](#)

[Unified Write Filter](#)

# UWF\_Volume.CommitFileDeletion

10/2/2018 • 2 minutes to read • [Edit Online](#)

Deletes the specified file and commits the deletion to the physical volume.

## Syntax

```
UInt32 CommitFileDeletion(  
    string FileName  
)
```

## Parameters

*FileName* [in] A string that contains the path of the file to delete, but does not include the drive letter or volume name. For example: "\users\test.dat".

## Return Value

Returns an HRESULT value that indicates [WMI status](#) or a [WMI error constant](#).

## Remarks

The *FileName* must contain the name of a file that exists on the physical volume. The **CommitFileDeletion** method cannot delete a file that does not exist.

You must use an administrator account to call this method.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[UWF\\_Volume](#)

[Unified Write Filter](#)

# UWF\_Volume.FindExclusion

10/2/2018 • 2 minutes to read • [Edit Online](#)

Checks if a specific file or folder is in the exclusion list for a volume protected by Unified Write Filter (UWF).

## Syntax

```
UInt32 FindExclusion (
    [in] string FileName,
    [out] boolean bFound
);
```

## Parameters

*FileName* [in] A string that contains the full path of the file or folder relative to the volume.

*bFound* [out] Indicates if *FileName* is in the file exclusion list for the volume.

## Return Value

Returns an HRESULT value that indicates [WMI status](#) or a [WMI error constant](#).

## Remarks

**FindExclusion** sets *bFound* to **true** only for file and folder exclusions that have been explicitly added to the exclusion list. Files and subfolders that are in an excluded folder are not identified as excluded by **FindExclusion**, unless they have been explicitly excluded.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[UWF\\_Volume](#)

[Unified Write Filter](#)

# UWF\_Volume.GetExclusions

10/2/2018 • 2 minutes to read • [Edit Online](#)

Gets a list of all file exclusions for a Unified Write Filter (UWF) protected volume.

## Syntax

```
UInt32 GetExclusions(  
    [out, EmbeddedInstance("UWF_ExcludedFile")] string ExcludedFiles[]  
)
```

## Parameters

### *ExcludedFiles*

[out] An array of [UWF\\_ExcludedFile](#) objects that represent the files and folders that are excluded from UWF filtering for a volume.

## Return Value

Returns an HRESULT value that indicates [WMI status](#) or a [WMI error constant](#).

## Remarks

If **GetExclusions** does not find any files or folders in the file exclusion list for the volume, **GetExclusions** sets the *ExcludedFiles* parameter to null.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[UWF\\_Volume](#)

[Unified Write Filter](#)

# UWF\_Volume.Protect

10/2/2018 • 2 minutes to read • [Edit Online](#)

Enables Unified Write Filter (UWF) to protect the volume after the next system restart, if UWF is enabled after the restart.

## Syntax

```
UInt32 Protect();
```

## Parameters

None.

## Return Value

Returns an HRESULT value that indicates [WMI status](#) or a [WMI error constant](#).

## Remarks

UWF starts protecting the volume after the next device restart in which UWF is enabled.

This method does not enable UWF if it is disabled; you must explicitly enable UWF for the next session to start volume protection.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[UWF\\_Volume](#)

[Unified Write Filter](#)

# UWF\_Volume.RemoveAllExclusions

10/2/2018 • 2 minutes to read • [Edit Online](#)

Removes all files and folders from the file exclusion list for a volume protected by Unified Write Filter (UWF).

## Syntax

```
UInt32 RemoveAllExclusions();
```

## Parameters

None.

## Return Value

Returns an HRESULT value that indicates [WMI status](#) or a [WMI errorj constant](#).

## Remarks

This command does not remove registry exclusions.

You must use an administrator account to remove file or folder exclusions, and you must restart the device for this change to take effect.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[UWF\\_Volume](#)

[Unified Write Filter](#)

# UWF\_Volume.RemoveExclusion

10/2/2018 • 2 minutes to read • [Edit Online](#)

Removes a specific file or folder from the file exclusion list for a volume protected by Unified Write Filter (UWF).

## Syntax

```
UInt32 RemoveExclusion(  
    string FileName  
)
```

## Parameters

*FileName* A string that contains the full path of the file or folder relative to the volume.

## Return Value

Returns an HRESULT value that indicates [WMI status](#) or a [WMI error constant](#).

## Remarks

You must use an administrator account to remove file or folder exclusions, and you must restart the device for this change to take effect.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[UWF\\_Volume](#)

[Unified Write Filter](#)

# UWF\_Volume.SetBindByDriveLetter

10/2/2018 • 2 minutes to read • [Edit Online](#)

Sets the **BindByDriveLetter** property, which indicates if the Unified Write Filter (UWF) volume is bound to the physical volume by drive letter or volume name.

## Syntax

```
UInt32 SetBindByDriveLetter(  
    boolean bBindByDriveLetter  
) ;
```

## Parameters

*bBindByDriveLetter* A Boolean value that indicates the type of binding to use. The **BindByDriveLetter** property is set to this value.

VALUE	DESCRIPTION
<b>true</b>	Binds the UWF volume by the drive letter ( <i>loose binding</i> ).
<b>false</b>	Binds the UWF volume by the volume name ( <i>tight binding</i> ).

## Return Value

Returns an HRESULT value that indicates [WMI status](#) or a [WMI error](#).

## Remarks

Binding by volume name is considered more reliable than binding by drive letter, since drive letters can change for a volume if devices are added or removed.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

UWF\_Volume

Unified Write Filter

# UWF\_Volume.Unprotect

10/2/2018 • 2 minutes to read • [Edit Online](#)

Disables UWF protection of the volume after the next system restart.

## Syntax

```
UInt32 Unprotect();
```

## Parameters

None.

## Return Value

Returns an HRESULT value that indicates [WMI status](#) or a [WMI error constant](#).

## Remarks

Unprotecting the volume does not remove the [UWF\\_Volume](#) entry or any configuration settings from the UWF configuration registry. This means that you can unprotect a volume, and then protect it again later, while keeping any file exclusions or volume configurations that you have defined.

## Requirements

WINDOWS EDITION	SUPPORTED
Windows 10 Home	No
Windows 10 Pro	No
Windows 10 Enterprise	Yes
Windows 10 Education	Yes

## Related topics

[UWF\\_Volume](#)

[Unified Write Filter](#)

# uwfmgr.exe

10/8/2018 • 6 minutes to read • [Edit Online](#)

The UWFMgr tool can be used at the command-line or in PowerShell to configure and retrieve settings for [Unified Write Filter \(UWF\)](#).

## **IMPORTANT**

Users with standard accounts can use commands that retrieve information, but only users who have administrator accounts can use commands that change the configuration settings.

## Syntax

```
uwfmgr.exe
    Help | ?
    Get-Config
    Filter
        Help | ?
        Enable
        Disable
        Reset-Settings
        Shutdown
        Restart
    Volume
        Help | ?
        Get-Config {<volume> | all}
        Protect {<volume> | all}
        Unprotect <volume>
    File
        Help | ?
        Get-Exclusions {<volume> | all}
        Add-Exclusion <file>
        Remove-Exclusion <file>
        Commit <file>
        Commit-Delete <file>
    Registry
        Help | ?
        Get-Exclusions
        Add-Exclusion <key>
        Remove-Exclusion <key>
        Commit <key> [<value>]
        Commit-Delete <key> [<value>]
    Overlay
        Help | ?
        Get-Config
        Get-AvailableSpace
        Get-Consumption
        Set-Size <size>
        Set-Type {RAM | DISK}
        Set-WarningThreshold <size>
        Set-CriticalThreshold <size>
        Set-Passthrough <on/off>
        Set-Persistent <on/off>
        Reset-PersistentState <on/off>
    Servicing
        Enable
        Disable
        Update-Windows
        Get-Config
        Help
```

## Location

**Uwfmgr** can be found under the %WINDIR%\System32\ folder.

## Command-line options and parameters

The following list describes the options and sub-options that are available to use in **uwfmgr.exe**, and it lists the corresponding WMI class or method for each command-line option and sub-option (if available).

- **Help | ?**
  - Displays command-line help for basic parameters for **uwfmgr.exe**.
- **Get-Config**
  - Displays UWF configuration settings for the current and next session.
- **Filter**

- Configures basic UWF settings.
- [UWF\\_Filter](#)
- *Enable*
  - Enables UWF protection for the next session after a system restart.
  - [UWF\\_Filter.Enable](#)
- *Disable*
  - Disables UWF protection for the next session after a system restart.
  - [UWF\\_Filter.Disable](#)
- *Reset-Settings*
  - Restores UWF settings to the original state.
 

If you added UWF to your image by using **Turn Windows features on or off** or by using DISM, the original state is the state of UWF settings when UWF was first enabled.

If you added UWF to your image by using SMI settings in an unattend file, the original state is the state of UWF settings when Windows 10 Enterprise was installed on the device.
  - [UWF\\_Filter.ResetSettings](#)
- *Shutdown*
  - Shuts down the device immediately, even if the overlay is full or near full. Administrator-level permissions are required to use this command.
  - [UWF\\_Filter.ShutdownSystem](#)
- *Restart*
  - Shuts down the device immediately and restarts, even if the overlay is full or near full. Administrator-level permissions are required to use this command.
  - [UWF\\_Filter.RestartSystem](#)

- **Volume**

- Configures settings for volumes protected by UWF. If the *<volume>* argument is needed, you can specify a drive letter (for example, `uwfmgr.exe volume protect c:`), or else you can specify all volumes (for example, `uwfmgr.exe volume get-config all`).
- [UWF\\_Volume](#)
- *Help | ?*
  - Displays command-line help for the `uwfmgr.exe volume` command.
- *Get-Config {<volume> | all}*
  - Displays configuration settings and file exclusions for the specified volume, or all volumes if **all** is specified. Displays information for both the current and the next session.
- [UWF\\_Volume](#)
- *Protect {<volume> | all}*
  - Adds the specified volume to the list of volumes that are protected by UWF. UWF starts protecting the volume after the next system restart if UWF filtering is enabled.
  - [UWF\\_Volume.Protect](#)
- *Unprotect <volume>*
  - Removes the specified volume from the list of volumes that are protected by UWF. UWF stops protecting the volume after the next system restart.
  - [UWF\\_Volume.Unprotect](#)

- **File**

- Configures file exclusion settings for UWF. If you use the *<file>* argument, it must be fully qualified, including the volume and path. **uwfmgr.exe** uses the volume specified in the *<file>* argument to determine which volume contains the file exclusion list for the file.
- [UWF\\_Volume](#)
- *Help | ?*

- Displays command-line help for the `uwmgr.exe file` command.
- *Get-Exclusions {<volume> | all}*
  - Displays all files and directories in the exclusion list for the specified volume (for example, `uwmgr.exe file Get-Exclusions C:`), or all volumes if **all** is specified. Displays information for both the current and the next session.
- [UWF\\_Volume.GetExclusions](#)
- *Add-Exclusion <file>*
  - Adds the specified file to the file exclusion list of the volume protected by UWF. UWF starts excluding the file from filtering after the next system restart.
- [UWF\\_Volume.AddExclusion](#)
- *Remove-Exclusion <file>*
  - Removes the specified file from the file exclusion list of the volume protected by UWF. UWF stops excluding the file from filtering after the next system restart.
- [UWF\\_Volume.RemoveExclusion](#)
- *Commit <file>*
  - Commits changes to a specified file to overlay for a UWF-protected volume. Administrator-level permissions are required to use this command.
- [UWF\\_Volume.CommitFile](#)
- *Commit-Delete <file>*
  - Deletes the specified file from both the overlay and the physical volume. Administrator-level permissions are required to use this command.
- [UWF\\_Volume.CommitFileDeletion](#)

- **Registry**

- Configures registry key exclusion settings for UWF.
- [UWF\\_RegistryFilter](#)
- *Help | ?*
  - Displays command-line help for the `uwmgr.exe registry` command.
- *Get-Exclusions*
  - Displays all registry keys in the registry exclusion list. Displays information for both the current and the next session.
- [UWF\\_RegistryFilter.GetExclusions](#)
- *Add-Exclusion<key>*
  - Adds the specified registry key to the registry exclusion list for UWF. UWF starts excluding the registry key from filtering after the next system restart.
- [UWF\\_RegistryFilter.AddExclusion](#)
- *Remove-Exclusion <key>*
  - Removes the specified registry key from the registry exclusion list for UWF. UWF stops excluding the registry key from filtering after the next system restart.
- [UWF\\_RegistryFilter.RemoveExclusion](#)
- *Commit <key> <value>*
  - Commits changes to the specified key and value. Administrator-level permissions are required to use this command.
- [UWF\\_RegistryFilter.CommitRegistry](#)
- *Commit-Delete <key> [<value>]*
  - Deletes the specified registry key and value and commits the deletion. Deletes all values and subkeys if the value is empty, and commits the deletion. Administrator-level permissions are required to use this command.
- [UWF\\_RegistryFilter.CommitRegistryDeletion](#)

- **Overlay**

- Configures settings for the UWF overlay.
- [UWF\\_Overlay](#) and [UWF\\_OverlayConfig](#)
- *Help | ?*
  - Displays command-line help for the `uwfmgr.exe overlay` command.
- *Get-Config*
  - Displays configuration settings for the UWF overlay. Displays information for both the current and the next session.
  - [UWF\\_Overlay](#) and [UWF\\_OverlayConfig](#)
- *Get-AvailableSpace*
  - Displays the amount of space remaining that is available for the UWF overlay.
  - [UWF\\_Overlay](#)
- *Get-Consumption*
  - Displays the amount of space currently used by the UWF overlay.
  - [UWF\\_Overlay](#)
- *Set-Size <size>*
  - Sets the maximum size of the UWF overlay, in megabytes, for the next session after a system restart.
  - [UWF\\_OverlayConfig.SetMaximumSize](#)
- *Set-Type {RAM | DISK}*
  - Sets the type of the overlay storage to RAM-based or disk-based. UWF must be disabled in the current session to set the overlay type to disk-based.
  - [UWF\\_OverlayConfig.SetType](#)
- *Set-WarningThreshold <size>*
  - Sets the overlay size, in megabytes, at which the driver issues warning notifications for the current session.
  - [UWF\\_Overlay.SetWarningThreshold](#)
- *Set-CriticalThreshold <size>*
  - Sets the overlay size, in megabytes, at which the driver issues critical notifications for the current session.
  - [UWF\\_Overlay.SetCriticalThreshold](#)
- *Set-Passthrough <on/off>*
  - Turns the [freespace passthrough](#) on or off, allowing UWF to use free space outside of the reserved space when available.
- *Set-Persistent <on/off>*
  - Sets the overlay as a [persistent overlay](#), allowing users to keep using their data after a reboot.
- *Reset-PersistentState <on/off>*
  - Clears a persistent overlay on the next boot (on/off).

- **Servicing**

- Configures settings for UWF servicing mode.
- [UWF\\_Servicing](#)
- *Enable*
  - Enables servicing mode in the next session after a restart. Administrator-level permissions are required to use this command.
  - [UWF\\_Servicing.Enable](#)
- *Disable*

- Disables UWF servicing mode in the next session after a restart. Administrator-level permissions are required to use this command.
- [UWF\\_Servicing.Disable](#)
- *Update-Windows*
  - Stand-alone command to apply Windows updates to a device. Called by the master servicing script that is called by the `uwmgr.exe servicing enable` command. We recommend that you use the `uwmgr.exe servicing enable` command to service your UWF-protected device whenever possible. Administrator-level permissions are required to use this command.
- [UWF\\_Servicing.UpdateWindows](#)
- *Get-Config*
  - Displays UWF servicing mode information for the current session and the next session.
- [UWF\\_Servicing](#)
- *Help*
  - Displays command-line help for the `uwmgr.exe servicing` command.

## Unsupported WMI methods

The following list contains the UWF WMI provider methods that are not currently supported by the **uwmgr.exe** tool:

- [UWF\\_Overlay.GetOverlayFiles](#)
- [UWF\\_RegistryFilter.FindExclusion](#)
- [UWF\\_Volume.FindExclusion](#)
- [UWF\\_Volume.RemoveAllExclusions](#)
- [UWF\\_Volume.SetBindByDriveLetter](#)

## Related topics

[Unified Write Filter](#)

# Windows System Image Manager Technical Reference

10/2/2018 • 2 minutes to read • [Edit Online](#)

Windows System Image Manager (Windows SIM) is the tool that you use to create unattended Windows Setup answer files.

Windows SIM is included with the Windows ADK. Download the Windows ADK [from this website](#).

You can create an answer file by using information from a Windows image (.wim) file and a catalog (.clg) file. Component settings are added to an appropriate configuration pass in the answer file. You can also add packages to be installed during Windows Setup. The following topics describe conceptual information about Windows SIM.

## IMPORTANT

If you experience problems creating catalog files by using Windows SIM, see [Windows Image Files and Catalog Files Overview](#). This topic contains information about known issues and workarounds for creating catalog files.

## In This Section

<a href="#">Windows System Image Manager Overview Topics</a>	Provides an overview of Windows SIM, the user interface, and important concepts for deploying Windows.
<a href="#">Windows System Image Manager How-to Topics</a>	Provides how-to instructions for using Windows SIM.
<a href="#">Windows System Image Manager Reference Topics</a>	Describes reference information for Windows SIM. This information includes components and settings and how Windows SIM works.

## Related topics

[Deployment Image Servicing and Management \(DISM\) Technical Reference](#)

# Windows System Image Manager Overview Topics

10/2/2018 • 2 minutes to read • [Edit Online](#)

Windows® System Image Manager (Windows SIM) is a GUI that you use to create and manage answer files. Answer files are .xml files that are used in Windows Setup, **Sysprep**, Deployment Image Servicing and Management (DISM), and other deployment tools to configure and customize the default Windows installation. You can access Windows SIM by searching for "Windows System Image Manager" on your computer.

You can use answer files to customize different aspects of Windows, including the default language settings, the partitions to create and format during installation, and the default settings for the Windows Internet Explorer® home page.

You can use Windows SIM to do the following:

- Create and manage answer files.
- Validate the settings of an answer file against a Windows image (.wim) file.
- View all of the configurable component settings in a Windows image.
- Create a configuration set that contains a complete set of portable folders with Setup files.
- Add third-party drivers, applications, or other packages to an answer file.

## In This Section

<a href="#">Windows System Image Manager Scenarios Overview</a>	Describes Windows SIM scenarios and when to use which scenario.
<a href="#">Windows System Image Manager User Interface Overview</a>	Describes the user interface of Windows SIM.
<a href="#">Windows Image Files and Catalog Files Overview</a>	Describes conceptual information about Windows image files and catalog (.clg) files.
<a href="#">Answer Files Overview</a>	Describes conceptual information about the structure of answer files.
<a href="#">Best Practices for Authoring Answer Files</a>	Describes recommendations to consider when you are creating and managing answer files.
<a href="#">Distribution Shares and Configuration Sets Overview</a>	Describes information about distribution shares, how they work, and when to create a configuration set.

## Related topics

[Windows System Image Manager How-to Topics](#)

[Windows System Image Manager Reference Topics](#)

# Windows System Image Manager Scenarios Overview

10/2/2018 • 5 minutes to read • [Edit Online](#)

Windows® System Image Manager (Windows SIM) creates and manages unattended Windows Setup answer files in a GUI.

Answer files are .xml files that are used during Windows Setup to configure and customize the default Windows installation.

For example, you can use Windows SIM to create an answer file that partitions and formats a disk before installing Windows. Windows SIM also changes the default setting for the Windows Internet Explorer® home page, and it configures Windows to boot to audit mode after installation. By modifying settings in the answer file, Windows SIM can also be used to install third-party applications, device drivers, language packs, and other updates.

## NOTE

Windows SIM does not modify the Windows image itself. You use Windows SIM only to create an answer file. During Windows Setup, the answer file applies the settings to the Windows installation. Windows SIM does not modify the settings in a Windows image (.wim) file.

## Common Windows SIM Scenarios

### Create a Catalog File for a Windows Image

Before you can create an answer file, you must create a catalog (.clg) file. Catalog files contain all of the configurable settings in a single Windows image and the current values of each setting.

We recommend that you use the 32-bit version of Windows SIM when you create your catalog files. The following table shows the architectures of Windows SIM and the supported Windows image architectures.

WINDOWS SIM ARCHITECTURE	CAN CREATE CATALOGS FOR WINDOWS IMAGES OF THE FOLLOWING ARCHITECTURE TYPES
x86 version of SIM	x86-based systems, x64-based systems, and Windows® RT ARM-based systems
x64 version of SIM	x64-based systems only

### Create a New Answer File for a Windows Image

You can use Windows SIM to create an answer file to be used during Windows Setup. You can view all of the components that are available in a Windows image, add component settings to your answer file, and choose when to apply a component setting by adding it to a particular configuration pass.

After you add component settings to an unattended answer file, you can view and customize the available settings for each component. For more information, see [Answer Files Overview](#).

### Edit an Existing Answer File

You can use Windows SIM to add components, packages, or other updates to an existing answer file. You can also

validate an existing answer file against a Windows image to ensure that the settings in that answer file can be applied to a specific Windows image. An answer file is typically associated with a specific Windows image. By using Windows SIM, you can open the Windows image, open an existing answer file, and then make changes to the answer file.

Windows SIM validates the component settings in the answer file against the settings that are available in the Windows image. For more information, see [How to Validate an Answer File](#).

### Add Device Drivers to an Answer File

You can add device drivers during Windows Setup by using Windows SIM. Windows Setup uses the following types of drivers:

- **In-box drivers.** Windows Setup handles in-box drivers the same way that it handles packages.
- **Out-of-box drivers.** By using Windows SIM, you can add out-of-box drivers (INF-based) during Windows Setup. Typically, these out-of-box drivers are processed during the **auditSystem** configuration pass. Your .inf-based out-of-box drivers must be in a distribution share subfolder that is called Out-of-Box Drivers. For more information, see [How to Manage Files and Folders in a Distribution Share](#).
- **In-box drivers that are installed with a Windows Installer file.** In-box drivers that require a Windows Installer file are added the same way that applications are added.

#### NOTE

By using the **Microsoft-Windows-PnpCustomizationsWinPE** component, you must add boot-critical device drivers that are required for installation during the **windowsPE** configuration pass. For more information, see [How to Add Device Drivers by Using Windows Setup](#). You can also use Deployment Image Servicing and Management (**DISM**) to add device drivers to an offline image. For more information, see [How to Add and Remove Drivers Offline](#).

### Add Applications or Drivers to an Answer File

You can use Windows SIM to add applications or drivers to be installed during Windows Setup by using a distribution share. You use a distribution share to store all applications, device drivers, scripts, or other resources that you make available during Windows Setup.

You can add more applications, scripts, and other binary files by using a data image. A data image is packaged in a way that is similar to a Windows image. By using the **DISM** tool (**DISM.exe**), you can capture a folder structure that contains the resources that you must add to Windows (or another partition on the computer) during Windows Setup. You can specify where the data image is applied by using the **DataImage** setting in the **Microsoft-Windows-Setup** component. For more information, see [How to Create a Data Image](#).

You can also use **\$OEM\$ Folders** folder structures to place binary files and other applications in specific locations during Windows Setup. Applications are added from distribution shares through subfolders in **\$OEM\$ Folders**. You must also add a **RunSynchronous** setting to the answer file to open the Windows Installer file or the .exe file that installs the application. For more information, see [How to Manage Files and Folders in a Distribution Share](#).

### Add Updates to a Windows Image Offline

Windows SIM enables the addition of offline updates to a Windows image. These updates include software updates, device drivers, language packs, and other packages, which Microsoft provides.

**DISM.exe** is the tool that you use, with or without an answer file, to apply packages to Windows. Any package installation, removal, or modification in the answer file is applied to the Windows image. For more information, see [How to Add or Remove Packages Offline](#).

Packages that exist in the **offlineServicing** configuration pass are applied to the offline Windows image. For more information, see [Windows Image Files and Catalog Files Overview](#).

### Create a Configuration Set

A configuration set is a subset of files that are available in a distribution share that is explicitly called in an answer file. When you create a configuration set, any files in a distribution share that are referenced in the answer file are saved to a specific folder. Paths to these files are updated in the answer file to point to the specific folder.

Configuration sets are smaller, more portable versions of a distribution share. A configuration set is ideal for installations that cannot access a distribution share. For more information, see [Distribution Shares and Configuration Sets Overview](#).

### **Import Packages to a Distribution Share**

Windows SIM imports packages that are not part of a Windows image file to an optional set of folders called a distribution share. You can then add packages to an answer file from the distribution share. To import a package to a distribution share, you must use the Windows SIM tool or the Component Platform Interface (CPI) APIs. For more information, see [Distribution Shares and Configuration Sets Overview](#).

You can also import a package directly into an answer file. The answer file includes a pointer to the path of the package.

## Related topics

[Windows Setup Technical Reference](#)

[Deployment Image Servicing and Management \(DISM\) Technical Reference](#)

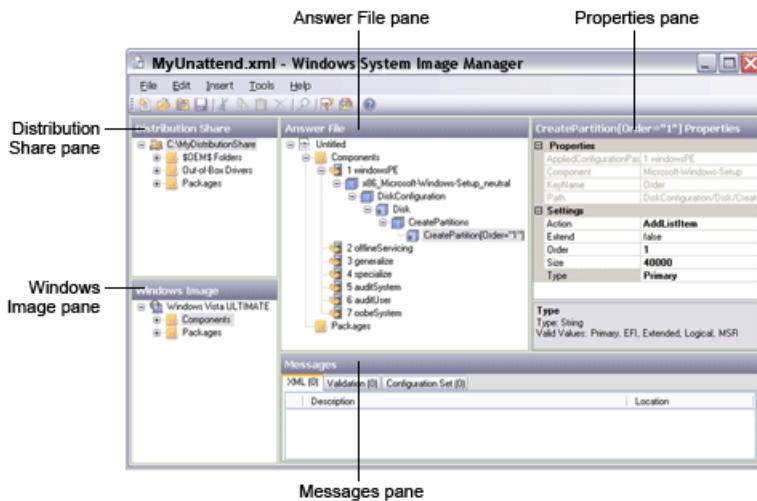
[System Preparation \(Sysprep\) Technical Reference](#)

# Windows System Image Manager User Interface Overview

10/2/2018 • 5 minutes to read • [Edit Online](#)

The Windows System Image Manager (Windows SIM) user interface contains a series of panes. You can use these panes to open Windows image (.wim) files, create unattended answer files, and then add components and packages to the respective configuration passes in an answer file.

The following screen shot illustrates the Windows SIM user interface.



## Windows SIM Panes

### Distribution Share Pane

The **Distribution Share** pane displays the currently open distribution share folder in tree view. You can select, create, explore, and close distribution share folders by selecting the top node and then right-clicking in the pane. You can add items in an open distribution share folder to an answer file by right-clicking the item. For more information, see [Distribution Shares and Configuration Sets Overview](#).

### Answer File Pane

The **Answer File** pane displays the Windows Setup configuration passes, the settings to apply in each pass, and the packages to install. You can open and change an existing answer file, validate the settings in an answer file against a Windows image, or create a new answer file. For more information, see [Answer Files Overview](#).

### Windows Image Pane

The **Windows Image** pane displays the currently open Windows image in tree view. When the tree is expanded, all of the components and packages for the image are visible and available to add to an answer file in the **Answer File** pane. For more information, see [Windows Image Files and Catalog Files Overview](#).

### Properties Pane

The **Properties** pane displays the properties and settings of a selected component or package. You can use the **Properties** pane to change the settings, and, in the case of packages, Windows feature selections. At the bottom of the **Properties** pane, Windows SIM displays the name of the setting and the associated **Microsoft® .NET** type. For more information, see [Component Settings and Properties Reference](#).

### Messages Pane

The **Messages** pane consists of three tabs: **XML**, **Validation**, and **Configuration Set**. Clicking a tab in the **Messages** pane displays the type of message, a description, and the location of the issue.

The types of messages that the **Messages** pane displays are informational. Messages appear on the **Configuration Set** tab if a configuration set has been created. For more information, see [Distribution Shares and Configuration Sets Overview](#).

## Windows SIM Menus

### File Menu

MENU COMMAND	DESCRIPTION
<b>New Answer File</b>	Creates a new answer file.
<b>Open Answer File</b>	Opens an existing answer file.
<b>Close Answer File</b>	Closes the currently open answer file.
<b>Save Answer File</b>	Saves the currently open answer file.
<b>Save Answer File As...</b>	Opens a dialog box to enable naming of the answer file with which you are currently working.
<b>Select Distribution Share</b>	Opens a valid distribution share folder.
<b>Close Distribution Share</b>	Closes the currently open distribution share folder.
<b>Select Windows Image</b>	Browses to and selects a Windows image file.
<b>Close Windows Image</b>	Closes the currently open Windows image file.
<b>Exit</b>	Closes Windows SIM.

### Edit Menu

MENU COMMAND	DESCRIPTION
<b>Cut</b>	Deletes the highlighted text or tree structure ( <b>Unattend</b> and <b>Properties</b> ).
<b>Copy</b>	Copies the highlighted text or tree structure (all panes).
<b>Paste</b>	Pastes text or tree structure ( <b>Unattend</b> and <b>Properties</b> ).

MENU COMMAND	DESCRIPTION
<b>Delete</b>	Deletes the currently selected item or text. (This command may be disabled if the item is not removable.)
<b>Find</b>	Opens a search dialog box to scan a Windows image and answer file, distribution share, or message for a specific item.
<b>Revert Change</b>	Reverts the most recent customization.
<b>Write Image Value</b>	Writes the value of the setting in the currently open Windows image to the answer file.
<b>Add to Answer File</b>	For components, a submenu opens that shows the available passes. The item and its children are added to the Unattend.xml answer file. Packages are automatically added to the packages section of the Unattend.xml answer file.

## Insert Menu

MENU COMMAND	DESCRIPTION
<b>Synchronous Command</b>	Adds a synchronous command to a configuration pass. You can select the <b>windowsPE</b> , <b>specialize</b> , <b>auditUser</b> , or <b>oobeSystem</b> configuration pass. After you select a configuration pass, a window opens so that you can specify the command line and the order of execution.
<b>Driver Path</b>	Adds a driver path to a configuration pass. You can use <b>Driver Path</b> to select the configuration pass in which to add the driver path. <b>Driver Path</b> then opens a window where you can select a file or folder.
<b>Package(s)</b>	Opens a window where you can browse to the location of a package. Then, it inserts a package from a file or folder into the currently open answer file.

## Tools Menu

MENU COMMAND	DESCRIPTION
<b>Hide Sensitive Data</b>	Stores local account passwords in an answer file as unreadable text. Domain passwords, product keys, and other sensitive data are not hidden.
<b>Validate Answer File</b>	Validates the XML and other settings in the answer file. Settings are validated against the currently open Windows image.

MENU COMMAND	DESCRIPTION
<b>Create Configuration Set</b>	Generates a new configuration set.
<b>Explore Distribution Share</b>	Opens a distribution share folder in Windows Explorer or File Explorer view.
<b>Create Distribution Share</b>	Creates a distribution share folder and subfolders.
<b>Import Package(s)</b>	Enables you to browse to a folder that contains a package, and then import it into the currently open distribution share. For more information, see <a href="#">Add Packages to a Distribution Share</a> .
<b>Create Catalog</b>	Generates a catalog file. For more information, see <a href="#">Open a Windows Image or Catalog File</a> .

## Help Menu

MENU COMMAND	DESCRIPTION
<b>Image Manager Help</b>	Displays the User's Guide.
<b>Unattended Reference</b>	Displays the Unattended Windows Setup Reference.
<b>About</b>	Displays version, copyright, and licensing information.

## Windows SIM Buttons

BUTTON NAME	FUNCTION
<b>New Answer File</b>	Creates a new answer file.
<b>Open Answer File</b>	Opens an existing answer file.
<b>Close Answer File</b>	Closes the currently selected answer file.
<b>Save Answer File</b>	Saves the currently open answer file.
<b>Cut, Copy, Paste, Delete</b>	Manipulates data.
<b>Find</b>	Enables you to search through a Windows image and answer file, through a distribution share, or within the <b>Messages</b> pane.

BUTTON NAME	FUNCTION
<b>Validate Answer File</b>	Validates the answer file against the settings in the opened catalog file.
<b>Create Configuration Set</b>	Creates a configuration set.
<b>Help Contents</b>	Displays the User's Guide.

## Related topics

[Windows System Image Manager Scenarios Overview](#)

[Windows System Image Manager Overview Topics](#)

# Windows Image Files and Catalog Files Overview

10/2/2018 • 4 minutes to read • [Edit Online](#)

Windows System Image Manager (Windows SIM) uses Windows image (**.wim**) files and catalog (**.clg**) files to display the available components and packages that can be added to an answer file (**Unattend.xml**). Windows images and catalog files contain configurable settings that you can modify after the component or package is added to an answer file.

## TIP

Install.wim is located in the **Sources** folder of your Windows Installation Media download. See [OEM deployment of Windows 10 for desktop editions](#) for steps to make and deploy Windows images.

You can open Windows SIM by searching your computer for "Windows System Image Manager".

## Supported architectures

Windows SIM can create catalog files for Windows images of the following architecture types

YOUR VERSION OF WINDOWS	WINDOWS IMAGES YOU CAN CREATE CATALOG FILES FROM
x86 version of Windows	x86-based systems, x64-based systems, and ARM-based systems
x64 version of Windows	x64-based systems only

Don't have an x86 PC handy?

- You can install the 32-bit version of Windows on a 64-bit PC.
- You can install Windows on a 32-bit virtual machine from a 64-bit PC.

## Windows Image Files

A Windows image file contains one or more compressed Windows images. Each Windows image in a Windows image file contains a list of all of the components, settings, and packages that are available with that Windows image.

### Limitations of Windows Image Files

The following list describes some of the limitations of using Windows image files:

- Only an account that has administrator permissions can open Windows image files.
- Only one user at a time can open Windows image files.
- Because Windows image files can contain one or more Windows images, they are frequently large. Some Windows image files can be several gigabytes in size.
- Because Windows images can be modified through different settings, using a Windows image file to create your answer file might cause you to apply altered default settings and configurations to a recaptured Windows image.

Because of these limitations, Windows SIM uses catalog files to create an answer file.

# Catalog Files

A catalog file is a binary file that only includes the settings and packages in a Windows image. Catalog files (.clg) are only used by Windows SIM and is not used by other deployment tools, nor is it required to install Windows. When Windows SIM creates a catalog file, it queries the Windows image for a list of all the settings and state of each setting in that image. Because the contents of a Windows image can change over time, you must re-create the catalog file whenever you update a Windows image.

Because only administrators can open Windows images, you must have administrator permissions on the system to create a catalog file.

When Windows SIM opens a Windows image file or catalog file, all of the configurable components and packages inside that image are displayed in the **Windows Image** pane. You can then add components and settings to an answer file.

## Contents of a Catalog File

A catalog file contains the following information:

- A list of component settings and current values
- Windows features and package states

## Benefits of Catalog Files

Catalog files have several advantages over Windows image files:

- The size of a catalog file can be less than 1 megabyte (MB), whereas the size of Windows image files can be several gigabytes. Also, catalog files are easier to copy to removable media or a network share.
- Multiple users can create answer files for a single catalog file at the same time, whereas only one person can open and access a Windows image file at any particular time.
- Non-administrators can create answer files for a catalog file. However, only administrators can open Windows image files.

## Troubleshooting

- **"The catalog file for Windows Image (image name) cannot be opened for the following reason:**

**Cannot find the catalog file associated with the Windows image (image name)**

**You must have a valid catalog file to continue. Do you want to create a catalog file?"**

*Fix:* Click **Yes** to create a catalog file. After you've created the catalog file, this message will no longer appear.

*What's going on:* This message usually shows up the first time you open a .wim file.

- **"Access denied"**

*Fix:* Copy the .wim file to a simple writable file location, like C:\Images, then try again.

*What's going on:* This message appears when you're creating a catalog file from a .wim file that's in a location that the system can't write to, like a DVD or secured network share.

- **"Catalog creation failed to complete. This 64-bit version of Windows SIM can only create catalogs for 64-bit Windows images. For a list of supported architecture types, see link below."**

*Fix:* Use an x86 version installation of Windows to create catalog files for x86 or ARM-based .wim files.

*What's going on:* Windows SIM can't create x86 or ARM catalog files from a 64-bit Windows installation. See [architectures](#).

## Related topics

[Open a Windows Image or Catalog File](#)

[Windows System Image Manager Overview Topics](#)

# Answer Files Overview

10/2/2018 • 2 minutes to read • [Edit Online](#)

An answer file is an XML-based file that contains setting definitions and values to use during Windows Setup. In an answer file, you specify various setup options. These options include how to partition disks, where to find the Windows image that will be installed, and which product key to apply. You can also specify values that apply to the Windows installation, such as names of user accounts and display settings. The answer file for Setup is typically called Unattend.xml.

Answer files that are created in Windows System Image Manager (Windows SIM) are associated with a particular Windows image. You can therefore validate the settings in the answer file to the settings in the Windows image. However, because any answer file can be used to install any Windows image, if there are settings in the answer file for components that are not in the Windows image, those settings are ignored. For information about how to create answer files, see [Best Practices for Authoring Answer Files](#).

## Sections of an Answer File

Settings in an answer file are organized into two sections, components and packages.

### Components

The components section of an answer file contains all the component settings that are applied during Windows Setup. Components are organized into various configuration passes: **windowsPE**, **offlineServicing**, **generalize**, **specialize**, **auditSystem**, **auditUser**, and **oobeSystem**. Each configuration pass represents a different phase of Windows Setup. Settings can be applied during one or more passes. If a setting can be applied in more than one configuration pass, you can select the pass in which to apply the setting.

For more information about the different components and settings that you can configure in an answer file, see the [Unattended Windows Setup Reference](#) (unattend.chm).

### Packages

Microsoft uses packages to distribute software updates, service packs, and language packs. Packages can also contain Windows features.

You can configure packages to be added to a Windows image or removed from a Windows image. You can also change the settings for features in a package.

The Windows Foundation Package, included in all Windows client and server images, includes Windows features. For example, Windows Media Player, Games, and Backup are all Windows features in the Windows Foundation Package.

Features are either enabled or disabled in Windows. If a Windows feature is enabled, the resources, executable files, and settings for that feature are available to users on the system. If a Windows feature is disabled, the package resources are not available, but the resources are not removed from the system.

Some Windows features may require that you install other features before you can enable the installed version of Windows. You must validate your answer file and add any required packages.

For example, you can disable the Windows Media Player feature to prevent end users from running Windows Media Player. However, because the package is disabled, those resources are not removed from the Windows image.

Packages in an answer file are applied to the Windows image during the **offlineServicing** configuration pass. You can also use Deployment Image Servicing and Management (DISM) to add packages to an offline Windows

image.

## Related topics

[Create or Open an Answer File](#)

[Windows System Image Manager Overview Topics](#)

# Best Practices for Authoring Answer Files

10/2/2018 • 6 minutes to read • [Edit Online](#)

We recommend the following best practices for creating answer files.

There are many ways in which you can use answer files. For more information about how to use an answer file with Windows Setup, see [Windows Setup Automation Overview](#). For more information about how to use an answer file with the **Sysprep** tool, [Using Answer Files with Sysprep](#). For more information about how to use an answer file with Deployment Image Servicing and Management (DISM), see [DISM Unattended Servicing Command-Line Options](#).

## Always Validate Answer Files in Windows SIM

The recommended way to author answer files is to create them in Windows System Image Manager (Windows SIM). However, if you use a manually authored answer file, you must validate the answer file in Windows SIM to verify that the answer file works.

Because available settings and default values can sometimes change, you must revalidate your answer file when you reuse it.

## Avoid Unnecessary Settings

You can introduce unnecessary settings by inserting a setting's parent node into the answer file.

Windows SIM does not create an empty setting in an answer file. Although empty settings are ignored during Windows Setup, empty strings can extend installation time. Therefore, as you author your answer file, remove any settings that are not required.

In general, it is best to expand down to the lowest level of a component and select only those elements that you intend to set. For the default value, you do not have to include the element unless it is a required element.

## Understand Configuration Passes

Configuration passes represent different phases of installation. Understanding what happens during each configuration pass is very important to creating answer files. For more information, review [Windows Setup Automation Overview](#) and [How Configuration Passes Work](#).

## Avoid Creating Empty Elements

Windows SIM supports creating empty elements in an answer file. By right-clicking a string setting type and then clicking **Write empty string**, you create an empty element in the answer file. However, some settings support empty elements, and some do not. In some cases, creating an empty element causes Windows Setup to fail. Before you create an empty element, see the component-setting documentation in the Windows® Unattended Setup Reference (Unattend.chm).

## Creating Architecture-Specific Sections for Each Configuration Pass

If you perform cross-platform deployments, do not duplicate components for different architecture types in a single answer file. If you have multiple components that apply to different architecture types in a single answer file, the installation program may apply settings in the components more than once, or it may apply the settings incorrectly.

For cross-platform deployments, you must create architecture-specific settings for each configuration pass in an answer file. For example, for a 32-bit preinstallation environment and a 64-bit destination computer, you must specify only x86-based components in the **windowsPE** configuration pass and only x64-based components in all other configuration passes.

For 64-bit answer files, the wow64 settings are the 32-bit versions of an app, for those apps that include both 32-bit and 64-bit modes.

## Improve Security for Answer Files

Answer files store sensitive data, including product keys, passwords, and other account information. You can help protect this sensitive data by following these best practices:

- **Restrict access to answer files.** Depending on your environment, you can change the access control lists (**ACLs**) or permissions on a file. Only approved accounts can access answer files.
- **Hide passwords.** To improve security in answer files, you can hide the passwords for local accounts by using Windows SIM. For more information, see [Hide Sensitive Data in an Answer File](#).
- **Delete the cached answer file.** During unattended Windows installation, answer files are cached to the computer. For each configuration pass, sensitive information such as domain passwords and product keys are deleted in the cached answer file. However, other information is still readable in the answer file. Before you deliver the computer to a customer, delete the cached answer file in **%WINDIR%\panther**.

### NOTE

Delete the answer file only if no settings will be processed during the **oobeSystem** configuration pass. The **oobeSystem** configuration pass is processed immediately before Out-Of-Box Experience (OOBE) starts. This is typically the first time that a customer turns on the computer. If you delete the answer file from this folder, those settings will not be processed.

## Do Not Overwrite Existing Files When You Are Using Data Images or \$OEM\$ Folders

When you add data, such as additional drivers or applications, do not overwrite Windows system files. Overwriting system files can corrupt your computer. For information about how to add drivers and applications, see [How to Create a Data Image](#) and [How to Manage Files and Folders in a Distribution Share](#).

## Use Separate Answer Files to Deploy to Multiple Architecture Types

Create separate answer files for each architecture type that you intend to deploy to. If a single answer file contains multiple components that apply to different architecture types, the component settings may be applied more than once or may be applied incorrectly.

## Use Multiple Answer Files for Specific Customizations

You can use multiple answer files (Unattend.xml) to create different sets of customizations that you can apply to your images at different times. For example, you can use a generic answer file that contains your branding and support information during Windows Setup. After installation finishes, when you run the **Sysprep** tool, you can apply a second answer file to add more customizations. When you must service your Windows image, you can use a different answer file with **DISM**.

For example, you can define your basic customizations in an answer file that you use with Windows Setup. After installation finishes, you can use an answer file with **Sysprep** or **DISM**. For example, if you want to keep all of the drivers that were added to the installation during a **generalize** process, you can create an answer file to use with

**Sysprep** that contains the **PersistAllDeviceInstalls** setting. You can apply an answer file by running the following command: **Sysprep /generalize /unattend:answerfile**.

For more information about how to use an answer file with Windows Setup, see [Windows Setup Command-Line Options](#).

For more information about how to use an answer file with **Sysprep**, see [Sysprep Command-Line Syntax](#).

For more information about how to use an answer file with DISM, see [DISM Unattended Servicing Command-Line Options](#).

## Use the Correct Mechanisms to Add Updates to a Windows Image

Use only the Microsoft-supported servicing mechanisms to update a Windows image.

Use **DISM** to update an offline Windows image. For more information, see [Service an Offline Image](#).

During installation, you can also configure the computer to automatically download updates from Windows Update.

### WARNING

Never overwrite Windows system files by using **\$OEM\$** Folders subfolders or data images.

If you have additional device drivers to add to a computer, add these drivers offline by using **DISM**. You can also include additional drivers in an unattended installation by using the **Microsoft-Windows-PnPCustomizationsNonWinPE** and **Microsoft-Windows-PnPCustomizationWinPE** components. For more information, see [How to Add and Remove Drivers Offline](#).

## Specify Language Settings

To change languages by using an answer file, use the **Microsoft-Windows-International-Core-WinPE** component. There are two components in which you can specify language settings:

- **Microsoft-Windows-International-Core-WinPE**. Language settings are applied during the **windowsPE** configuration pass.
- **Microsoft-Windows-International-Core**. Language settings are applied during the **specialize** or **oobeSystem** configuration pass.

Because some languages require a restart, we recommend that you configure your language settings during the **windowsPE** configuration pass because the computer will always restart. If you process language settings during the **specialize** or **oobeSystem** configuration pass, the computer might require an additional restart.

## Use the Sysprep/generalize Command with LocalAccounts to Change Account Information

You can use the **Sysprep** command with the **generalize** option and the **LocalAccounts** settings to change account information about an existing user account.

If you specify the settings in the following example in the **specialize** configuration pass, all the values of **NEWVALUE** will be changed. However, *MyAccount* will retain its security group memberships. *MyAccount* is considered to be the same account with a different display name, description, and password value.

```
<LocalAccount>
  <Name>MyAccount</Name>
  <DisplayName>NEWVALUE</DisplayName>
  <Description>NEWVALUE</Description>
  <Password>
    <PlainText>false</PlainText>
    <Value>NEWVALUEBASE64</Value>
  </Password>
</LocalAccount>
```

## Related topics

[Windows System Image Manager \(Windows SIM\) Technical Reference](#)

[Sysprep Overview](#)

[Windows Setup Technical Reference](#)

[Deployment Image Servicing and Management \(DISM\)](#)

# Distribution Shares and Configuration Sets Overview

10/2/2018 • 6 minutes to read • [Edit Online](#)

A distribution share is an optional set of folders that contain custom scripts, images, branding, applications, drivers, and other files. These files can be copied to Windows® during installation through an answer file (Unattend.xml).

During installation, Windows connects to the path of the server share by using the credentials that you specify in an answer file. Only the files that you specify in the answer file are copied to the Windows installation.

If you are installing Windows in an environment that does not have a network share or server share, you can copy the necessary files from the distribution share to a configuration set. A configuration set is a subset of the files in a distribution share. You can copy a configuration set to external storage, such as a USB flash drive or an external hard disk, to use during installation.

## Folders in a Distribution Share

When you create a distribution share by using Windows System Image Manager (Windows SIM), three folders are created automatically. The folders are named **\$OEM\$ Folders**, **Out-of-Box Drivers**, and **Packages**. If you create your own distribution share, it must contain at least one of these folders for Windows SIM to recognize it as a valid distribution share.

### \$OEM\$ Folders

You can use the **\$OEM\$ Folders** folder and subfolders only when you are creating configuration sets. You can use **\$OEM\$ Folders** to include logos for branding and to add applications and other files that customize the unattended installation.

As a general rule, to add new files and resources to Windows, use a data image. For more information, see [How to Create a Data Image](#).

For more information about how to use **\$OEM\$ Folders**, see [How to Manage Files and Folders in a Distribution Share](#).

#### IMPORTANT

Do not overwrite existing files that are carried and serviced by the operating system. Using **\$OEM\$ Folders** to update or overwrite these files can cause the operating system to behave unpredictably and cause serious issues.

The following table describes the support for **\$OEM\$ Folders** and its subfolders.

FOLDER	DEFINITION	SUPPORTED
<b>\$OEM\$ Folders</b>	Contains all supplemental folders and files for an automated or customized installation.	Yes
<b>\$OEM\$ Folders\Textmode</b>	Contains updated mass-storage drivers and hardware abstraction layer (HAL) files that the text-mode part of Setup requires.	No

FOLDER	DEFINITION	SUPPORTED
<b>\$OEM\$ Folders\$\$</b>	Contains files that Windows Setup copies to the %WINDIR% folder (for example, C:\Windows) during installation.	Yes
<b>\$OEM\$ Folders\$\$\Help</b>	Contains custom Help files that Windows Setup copies to the %WINDIR%\Help folder during installation.	No
<b>\$OEM\$ Folders\$\$\System32</b>	Contains files that Windows Setup copies to the %WINDIR%\System32 folder during installation.	Yes
<b>\$OEM\$ Folders\$1</b>	Represents the root of the drive on which you installed Windows (also called the boot partition), and contains files that Windows Setup copies to the boot partition during installation.	Yes
<b>\$OEM\$ Folders\$1&lt;/strong&gt;Pnpdrivers</b>	Contains new or updated Plug and Play (PnP) drivers. You specify the folder name in the Unattend.xml file for unattended installations. For example, you might name this folder \$OEM\$ Folders\$1\Pnpdrv\$.	Yes
<b>\$OEM\$ Folders\$1\SysPrep</b>	Contains files that are used for Sysprep-based installation.	No
<b>\$OEM\$ Folders\$Docs</b>	Contains files that Windows Setup copies to %DOCUMENTS_AND_SETTINGS% during installation.	No
<b>\$OEM\$ Folders\$Progs</b>	Contains programs that Windows Setup copies to the %PROGRAM_FILES% folder during installation.	No
<b>\$OEM\$ Folders\$Progs\Internet Explorer</b>	Contains the settings file to customize Windows Internet Explorer®.	No

FOLDER	DEFINITION	SUPPORTED
<b>\$OEM\$</b> <b>Folders&lt;/strong&gt;<i>drive_letter</i>&lt;em&gt;</b> <b>m&gt;subfolder</b>	A subfolder of the drive that contains files that Windows Setup copies to the subfolder during installation. Multiple instances of this type of folder can exist under the <b>\$OEM\$</b> <b>Folders&lt;/strong&gt;<i>drive_letter</i></b> <b>folder, for example, \$OEM\$</b> <b>Folders\D\MyFolder.</b>	Yes

## Out-of-Box Drivers

Drivers are a type of software that enables hardware or devices to function.

The **Out-of-Box Drivers** folder includes additional device drivers that you install during Windows Setup by using Windows SIM. Windows Setup uses the following types of drivers:

- **In-box drivers.** Windows Setup handles in-box drivers the same way that it handles packages.
- **Out-of-box drivers.** By using Windows SIM, you can add out-of-box device drivers that are based on .inf files. Typically, these out-of-box drivers are processed during the **auditSystem** configuration pass. Your .inf-based out-of-box drivers must be in a distribution-share subfolder that is called Out-of-Box Drivers. For more information, see [How to Manage Files and Folders in a Distribution Share](#).
- **In-box drivers that are installed via a .msi file.** In-box drivers that require a .msi file are added the same way that applications are added. > [!Note] > By using the **Microsoft-Windows-PnpCustomizationsWinPE** component, you must add boot-critical device drivers that are required for installation during the **windowsPE** configuration pass. For more information, see [How to Add Device Drivers by Using Windows Setup](#). You can also add device drivers to an offline image by using Deployment Image Servicing and Management (**DISM**). For more information, see [How to Add and Remove Drivers Offline](#).

## Packages

The **Packages** folder is a location for Windows software updates. Package types include service packs, security updates, language packs, and other packages that Microsoft issues. You must use Windows SIM to import packages to a distribution share. After a package is imported and available in the **Distribution Share** pane, you can add the package to the answer file. For more information, see [How to Add Packages to a Distribution Share](#).

# Configuration Sets

After an answer file (**Unattend.xml**) has been validated and saved, you can create a configuration set. A configuration set is a subset of a distribution share that you can create by using Windows SIM. Configuration sets are useful when a network share is not available. You can store configuration sets on removable media and use them in the field. Creating a configuration set exports binaries that are referenced in the answer file and puts them together in a self-contained file set that can be accessed from the Unattend.xml file.

## Contents of a Configuration Set

A configuration set contains a complete collection of files, drivers, applications, patches, and answer files that are used to customize Windows installations. A configuration set contains all the required binaries, which are packaged with an associated answer file (**Unattend.xml**).

## Benefits of Configuration Sets

Using configuration sets for unattended installations provides the following benefits:

- A configuration set is a smaller and more portable version of a distribution share, which can have a size of several gigabytes. You can use configuration sets to install Windows operating systems while you are in the

field.

- Configuration sets are completely self-contained and have no references outside the file set.
- You can duplicate a configuration set and then edit it for each computer model that you manufacture and release.

#### IMPORTANT

If a configuration set is used during Windows Setup, all the contents at the root of the media where the answer file exists are copied to the Windows installation. Having many files and folders at the same level as the answer file might slow down installation. In some cases, you might run out of disk space.

## Security Considerations for Distribution Shares and Configuration Sets

Your distribution shares and configuration sets contain private data. The following are recommendations for improving security for distribution shares and configuration sets:

- Restrict access to the contents of distribution shares. Depending on your environment, you can change the access control lists (**ACLs**) or permissions on a distribution share. Only approved accounts should have access to distribution shares.
- Keep applications and device drivers updated with the latest fixes and patches.

## Related topics

[How to Create or Open a Distribution Share](#)

[How to Manage Files and Folders in a Distribution Share](#)

[How to Add Packages to a Distribution Share](#)

[Windows SIM Technical Reference](#)

# Windows System Image Manager How-to Topics

10/2/2018 • 2 minutes to read • [Edit Online](#)

The following topics describe how to use Windows System Image Manager (Windows SIM).

## In This Section

<a href="#">Open a Windows Image or Catalog File</a>	Open a Windows image in Windows SIM and create a catalog (.clg) file.
<a href="#">Create or Open an Answer File</a>	Use Windows SIM to open or create an answer file.
<a href="#">Configure Components and Settings in an Answer File</a>	Use Windows SIM to add a component or change a setting in an answer file.
<a href="#">Validate an Answer File</a>	Use Windows SIM to validate an answer file against a Windows image.
<a href="#">Hide Sensitive Data in an Answer File</a>	Hide sensitive data, such as local account passwords, in an answer file.
<a href="#">Add a Device Driver Path to an Answer File</a>	Add a path for a device driver to an answer file.
<a href="#">Add a Package to an Answer File</a>	Add a package, such as a language pack, to an answer file.
<a href="#">Add a Custom Command to an Answer File</a>	Create a custom command to run during installation.
<a href="#">Find a Component, Setting, or Package in Windows SIM</a>	Use Windows SIM to find a component, setting, or package.
<a href="#">Create a Configuration Set</a>	Create a configuration set, which is a container of files, scripts, and other items that are referenced in an answer file.
<a href="#">Create or Open a Distribution Share</a>	Use Windows SIM to create or open a distribution share.
<a href="#">Manage Files and Folders in a Distribution Share</a>	Add files and folders, or remove files and folders, from a distribution share.

## [Add Packages to a Distribution Share](#)

Use Windows SIM to add a package, such as a language pack, to a distribution share.

## Related topics

[Windows Deployment Options](#)

[Windows System Image Manager Technical Reference](#)

# Open a Windows Image or Catalog File

10/2/2018 • 2 minutes to read • [Edit Online](#)

When you open a Windows® image (.wim) file in Windows System Image Manager (Windows SIM), a catalog (.clg) file is automatically created. If a catalog file already exists, Windows SIM re-creates the catalog file based on the contents of the Windows image that you select. When a catalog file is created, it queries the Windows image for a listing of all the settings in that image.

To create an answer file, you must first open a Windows image file or catalog file in Windows SIM. For more information about Windows image files and catalog files, see [Windows Image Files and Catalog Files Overview](#).

## Open a Windows image file or catalog file

1. Copy a previously created catalog file (.clg) to the technician computer or copy your customized Windows image file (install.wim) to the technician computer.

### TIP

Install.wim is located in the **Sources** folder of your Windows Installation Media download. See [OEM deployment of Windows 10 for desktop editions](#) for steps to make and deploy Windows images.

2. On the technician computer, open Windows SIM. One way to do this is to search for "Windows System Image Manager".
3. On the **File** menu, click **Select Windows Image**.
4. In the **Select a Windows Image** dialog box, select the file type in the **Files of type** drop-down list, and then browse to a Windows image file or catalog file. If you open a Windows image file, Windows SIM will automatically create a catalog of that Windows image.
5. If there is more than one type of Windows image in the file, select a specific Windows image in the **Select an Image** box. The Windows image file or catalog file appears in the **Windows Image** pane.
6. Click **Open**. If you have not previously opened that Windows image file or have not refreshed the catalog file recently, Windows SIM prompts you to create or re-create the catalog file.

## Create a catalog file

1. Open Windows SIM.
2. On the **Tools** menu, click **Create Catalog**. The **Open a Windows Image** dialog box opens.
3. Select a Windows image file, and then click **Open**. If you select a Windows image file that has more than one Windows image, the **Select an Image** dialog box opens.
4. Click to select an image type (for example, **Fabrikam Custom Image 1**), and then click **OK**. The catalog file is created in the same directory as the Windows image file that you selected.

## Troubleshooting

If Windows SIM does not create the catalog file, try the following steps:

- Make sure you are using the Windows 8.1 version of the Windows Assessment and Deployment Kit (Windows ADK).
- To create a catalog file for 32-bit or ARM-based PCs, use a 32-bit PC.

- Make sure the Windows base-image file (Install.wim) is in a folder that has read-write privileges, such as a USB flash drive or on your hard drive.

**IMPORTANT**

Windows SIM cannot create catalog files for some Windows images of different architecture types. For information about the support of cross-platform catalog creation, see [Windows Image Files and Catalog Files Overview](#).

## Related topics

[Windows System Image Manager How-to Topics](#)

# Create or Open an Answer File

10/2/2018 • 2 minutes to read • [Edit Online](#)

The following procedure describes how to create a new answer file or open an existing answer file by using Windows® System Image Manager (Windows SIM).

After you create or open an answer file, you can add settings and packages to it. For more information, see [Configure Components and Settings in an Answer File](#) and [Add a Package to an Answer File](#).

## Create an answer file

1. Open Windows SIM.
2. Open a Windows image. For more information, see [Open a Windows Image or Catalog File](#).
3. In the **Answer File** pane, select the top node, and then right-click to select **New Answer File**.

## Open an existing answer file

1. Open Windows SIM.
2. Right-click the **Answer File** pane, and then click **Open Answer File**. The **Open** dialog box appears.
3. Browse to the existing answer file, and then click **Open**.

The answer file appears in the **Answer File** pane.

### NOTE

The Windows image file that generated the answer file also opens if it is still in its original location.

## Troubleshooting

In some cases, Windows SIM might display validation errors when opening an existing answer file. If this happens, try the following options:

- Problems with individual settings in an answer file appear in the Messages pane. Use this information to identify and address the problem.
- If the answer file opens, but all the settings are listed as “does not exist” in the Messages pane, then the file you’re using might be for the wrong PC architecture – for instance, your original answer file might be based on x86, and your Windows catalog file is amd64. To fix this, you can find and replace `processorArchitecture="x86"` for `processorArchitecture="amd64"` and re-open the file.
- If Windows SIM can’t open the file at all, this often means there’s some malformed XML in the answer file. You can often narrow down the problem by cutting out sections of the answer file, one large block at a time, and trying again to re-open the file.

## Related topics

[Windows System Image Manager How-to Topics](#)

[Configure Components and Settings in an Answer File](#)

[Validate an Answer File](#)

[Hide Sensitive Data in an Answer File](#)

[Add a Device Driver Path to an Answer File](#)

[Add a Package to an Answer File](#)

[Add a Custom Command to an Answer File](#)

[Find a Component, Setting, or Package in Windows SIM](#)

# Configure Components and Settings in an Answer File

10/2/2018 • 2 minutes to read • [Edit Online](#)

The following procedure describes how to do the following:

- Add a component to an answer file
- Customize a setting value in an answer file
- Add a list item to an answer file

List items are settings that can be declared multiple times, and each declaration contains different values. For example, the Windows Internet Explorer **FavoritesList** setting is used to specify multiple Favorites links. For each new Favorites link that you want to add to Internet Explorer, you add a new **FavoritesList** setting.

## Add a component to an answer file

1. Open Windows System Image Manager (Windows SIM).
2. Open a Windows image. For more information, see [Open a Windows Image or Catalog File](#).
3. Create or open an answer file. For more information, see [Create or Open an Answer File](#).
4. In the **Windows Image** pane, locate the component or package that you want to add to the answer file.
5. Right-click the component, and then select a configuration pass.

The component is added to the answer file in the specified configuration pass.

### NOTE

To search the entire Windows image (.wim) file, press Ctrl+F.

## Customize a setting value in an answer file

1. Open Windows SIM.
2. Open a Windows image. For more information, see [Open a Windows Image or Catalog File](#).
3. Create or open an answer file. For more information, see [Create or Open an Answer File](#).
4. In the **Answer File** pane, find the configuration pass that contains the component for the setting that you want to change.
5. Select the component or package that contains the setting that you want to change.
6. In the **Settings** section of the **Properties** pane, change the value of the setting to update it. Depending on the type of setting, you can enter a new setting or select from a drop-down list of possible settings.

For more information about the various settings and properties, see [Component Settings and Properties Reference](#).

## Add a list item to an answer file

1. Open Windows SIM.
2. Open a Windows image. For more information, see [Open a Windows Image or Catalog File](#).
3. Create or open an answer file. For more information, see [Create or Open an Answer File](#).
4. In the **Windows Image** pane, add the component or setting to the answer file.

5. In the **Answer File** pane, right-click the list item, and then click **Insert**. Windows SIM inserts a new list-item type. For example, right-click the **DriverPaths** setting, and then click **Insert new PathAndCredentials**. A new **PathAndCredentials** setting is added to the answer file.
6. To add more list items, repeat the previous step.

**IMPORTANT**

Each list item should contain a unique **Key** value to differentiate the list item from other list items of the same type.

## Related topics

[Windows System Image Manager How-to Topics](#)

[Create or Open an Answer File](#)

[Validate an Answer File](#)

[Hide Sensitive Data in an Answer File](#)

[Add a Device Driver Path to an Answer File](#)

[Add a Package to an Answer File](#)

[Add a Custom Command to an Answer File](#)

[Find a Component, Setting, or Package in Windows SIM](#)

# Validate an Answer File

10/2/2018 • 2 minutes to read • [Edit Online](#)

Before you can save an answer file, you must validate the settings. After you successfully validate an answer file, you can apply all the setting values in the answer file to the Windows® image.

1. Open Windows System Image Manager (Windows SIM).
2. Open a Windows image. For more information, see [Open a Windows Image or Catalog File](#).
3. Create or open an answer file. For more information, see [Create or Open an Answer File](#).
4. On the **Tools** menu, click **Validate Answer File**.

Windows SIM compares the setting values in the answer file with the available settings in the Windows image.

- If the answer passes validation, a message appears in the **Messages** pane on the **Validation** tab. This message verifies that no warnings or errors occurred in the answer file. Otherwise, error messages appear in the same location.
- If an error occurs, double-click the error in the **Messages** pane to browse to the setting.
- If no modifications have been made to component settings, the values of the component settings are not saved in the answer file.

## Related topics

[Windows System Image Manager How-to Topics](#)

[Create or Open an Answer File](#)

[Configure Components and Settings in an Answer File](#)

[Hide Sensitive Data in an Answer File](#)

[Add a Device Driver Path to an Answer File](#)

[Add a Package to an Answer File](#)

[Add a Custom Command to an Answer File](#)

[Find a Component, Setting, or Package in Windows SIM](#)

# Hide Sensitive Data in an Answer File

10/2/2018 • 2 minutes to read • [Edit Online](#)

You can use Windows® System Image Manager (Windows SIM) to hide the password for the administrator account, and for any other user accounts on the local system, in an answer file. Hiding passwords in an answer file prevents users from reading the answer file and identifying passwords for local accounts.

The settings that you can hide include the following:

- **Microsoft-Windows-Shell-Setup | AutoLogon | Password**
- **Microsoft-Windows-Shell-Setup | UserAccounts | AdministratorPassword**
- **Microsoft-Windows-Shell-Setup | UserAccounts | LocalAccounts | LocalAccount | Password**

This option only hides the passwords in an answer file. It does not provide encryption or other security benefits. Consider answer files as sensitive data and be careful about authorizing access to your answer files.

## NOTE

You can hide only local account passwords in an answer file. Domain passwords, product keys, and other sensitive data may still be available as clear text in an answer file.

To hide account passwords in an answer file:

1. Open Windows SIM.
2. Open a Windows image. For more information, see [Open a Windows Image or Catalog File](#).
3. Create or open an answer file. For more information, see [Create or Open an Answer File](#).
4. Add one of the following password settings to your answer file:
  - **Microsoft-Windows-Shell-Setup | AutoLogon | Password**
  - **Microsoft-Windows-Shell-Setup | UserAccounts | AdministratorPassword**
  - **Microsoft-Windows-Shell-Setup | UserAccounts | LocalAccounts | LocalAccount | Password**
5. Add a value to one or more of the password settings. The component is added to the answer file in the specified configuration pass.
6. On the **Tools** menu, click **Hide Sensitive Data**. This makes sure that when the answer file is saved, the password information will be hidden.
7. Save the answer file and close Windows SIM. The answer file will resemble the following example.

```
<component name="Microsoft-Windows-Shell-Setup" processorArchitecture="x86"
publicKeyToken="31bf3856ad364e35" language="neutral" versionScope="nonSxS"
xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <UserAccounts>
        <LocalAccounts>
            <LocalAccount wcm:action="add">
                <Password>
                    <Value>UABhAHMACwB3AG8AcgBkADEAMgAzADQANGBQAGEAcwBzAhcAbwByAGQA</Value>
                    <PlainText>false</PlainText>
                </Password>
                <Description>MyAccountName</Description>
                <DisplayName>MyAccountName</DisplayName>
                <Group>FabrikamGroup</Group>
                <Name>MyAccountName</Name>
            </LocalAccount>
        </LocalAccounts>
    </UserAccounts>
</component>
```

#### NOTE

Windows SIM adds the **PlainText** element to the answer file. This element is used during Windows Setup to indicate whether or not the password is in plain text.

## Related topics

[Windows System Image Manager How-to Topics](#)

[Create or Open an Answer File](#)

[Configure Components and Settings in an Answer File](#)

[Validate an Answer File](#)

[Add a Device Driver Path to an Answer File](#)

[Add a Package to an Answer File](#)

[Add a Custom Command to an Answer File](#)

[Find a Component, Setting, or Package in Windows SIM](#)

# Add a Device Driver Path to an Answer File

10/2/2018 • 2 minutes to read • [Edit Online](#)

The following procedure describes how to add a device driver path to an answer file by using Windows® System Image Manager (Windows SIM).

This device-driver path is used to process additional out-of-box device drivers during Windows Setup. Out-of-box device drivers can be copied to a Windows image during the **windowsPE** configuration pass. In this configuration pass, you can add boot-critical drivers to a Windows image before that image is installed.

## NOTE

You can add more drivers to the Windows installation during the **auditSystem** configuration pass.

When you select a driver path, you select a folder that contains one or more .inf drivers. The folder path is added to the answer file and, during an unattended installation, is referenced to find all drivers in the path and install them.

You can add only .inf drivers to a Windows image by using this procedure. You must install drivers that are packaged as a .exe file or other file types on a running Windows operating system.

To add a device-driver path to an answer file:

1. Open Windows SIM.
2. Create or open an answer file. For more information, see [Create or Open an Answer File](#).
3. On the **Insert** menu, click **Driver Path**.
4. Select the configuration pass in which you want to install the driver. This can be the **windowsPE** or the **auditSystem** configuration pass.

## NOTE

Adding a driver to the **auditSystem** configuration pass processes the driver during Audit mode only.

5. The **Browse for Folder** dialog box appears. Select the driver path that you want to add to the answer file, and then click **OK**. The driver path is added to the answer file under the configuration pass that you selected. Depending on the configuration pass that you selected, the driver path is included as a list item to one of the following components:

- **Microsoft-Windows-PnpCustomizationsWinPE** for the **windowsPE** configuration pass
- **Microsoft-Windows-PnpCustomizationsNonWinPE** for the **auditSystem** configuration pass

## NOTE

You can also drag drivers from an **Out-of-Box Drivers** folder in the **Distribution Share** pane to either the **windowsPE** or **auditSystem** configuration pass in the **Answer File** pane. Or, right-click to add it.

## Related topics

[Windows System Image Manager How-to Topics](#)

[Create or Open an Answer File](#)

[Configure Components and Settings in an Answer File](#)

[Validate an Answer File](#)

[Hide Sensitive Data in an Answer File](#)

[Add a Package to an Answer File](#)

[Add a Custom Command to an Answer File](#)

[Find a Component, Setting, or Package in Windows SIM](#)

# Add a Package to an Answer File

10/2/2018 • 2 minutes to read • [Edit Online](#)

The following procedures describe how to add a package to an answer file by using Windows® System Image Manager (Windows SIM).

You can add a package to an answer file by using one of three options: from the **Insert** menu, from the **Windows Image** pane, or from an open distribution-share folder in the **Distribution Share** pane.

## Add a package from the Insert menu

1. Open Windows SIM.
2. Create or open an answer file. For more information, see [Create or Open an Answer File](#).
3. Open a distribution share. For more information, see [Create or Open a Distribution Share](#).
4. On the **Insert** menu, click **Package(s)**. The **Select Package(s) to Insert** window opens.
5. Browse to the package that you want to add, and then click **Open**.
6. In the **Properties** pane, under **Settings**, select one of the following values for **Action: Install, Remove, Configure, or Stage**.

## Add a package from the Windows Image pane

1. Open Windows SIM.
2. Create or open an answer file. For more information, see [Create or Open an Answer File](#).
3. Open a distribution share. For more information, see [Create or Open a Distribution Share](#).
4. In the **Windows Image** pane, select the **Packages** node. Expand the node, right-click the package that you want to add to the answer file, and then click **Add to Answer File**.
5. In the **Properties** pane, under **Settings**, choose one of the following values for **Action: Install, Remove, Configure, or Stage**.

## Add a package from a distribution share

1. Open Windows SIM.
2. Create or open an answer file. For more information, see [Create or Open an Answer File](#).
3. Open a distribution share. For more information, see [Create or Open a Distribution Share](#).
4. Right-click the package that you want to add to the distribution share packages directory, and then click **Add to Answer File**. The package is added to the answer file in the packages section.
5. In the **Properties** pane, under **Settings**, choose one of the following values for **Action: Install, Remove, Configure, or Stage**.

## Related topics

[Windows System Image Manager How-to Topics](#)

[Create or Open an Answer File](#)

[Configure Components and Settings in an Answer File](#)

[Validate an Answer File](#)

[Hide Sensitive Data in an Answer File](#)

[Add a Device Driver Path to an Answer File](#)

[Add a Custom Command to an Answer File](#)

[Find a Component, Setting, or Package in Windows SIM](#)

# Add a Custom Command to an Answer File

10/2/2018 • 2 minutes to read • [Edit Online](#)

The following procedure describes how to configure a custom command to run automatically during Windows Setup.

1. Open Windows System Image Manager (Windows SIM).
2. Create or open an answer file. For more information, see [Create or Open an Answer File](#).
3. On the **Insert** menu, point to **Synchronous Command**, and then click a configuration pass on the submenu. The **Create Synchronous Command** dialog box opens.
4. In the **Enter command line** box, type the command-line syntax. In the **Order** box, select the order of the commands that will run, and then click **OK**. The command is added to the answer file in the selected configuration pass, as follows:
  - Commands that are added to the **1 windowsPE** configuration pass appear in the setting **Microsoft-  
Windows-Setup\RunSynchronous**.
  - Commands that are added to the **4 specialize** or **6 auditUser passes** configuration pass appear in the setting **Microsoft-  
Windows-Deployment\RunSynchronous**.
  - Commands that are added to the **7 oobeSystem** configuration pass appear in the setting **Microsoft-  
Windows-Shell-Setup\FirstLogonCommands**.

## NOTE

If you create a user account that does not include administrative rights, commands that are added to the **7  
oobeSystem** configuration pass may not be run. Details are as follows:

- If User Account Control is enabled, a dialog box appears when that user logs on for the first time. The dialog box provides an option to allow an administrator to apply the commands. If the user clicks **Cancel**, these commands are not run.
- If User Account Control is disabled, these commands are not run.

## Related topics

[Windows System Image Manager How-to Topics](#)

[Create or Open an Answer File](#)

[Configure Components and Settings in an Answer File](#)

[Validate an Answer File](#)

[Hide Sensitive Data in an Answer File](#)

[Add a Device Driver Path to an Answer File](#)

[Add a Package to an Answer File](#)

[Find a Component, Setting, or Package in Windows SIM](#)

# Find a Component, Setting, or Package in Windows SIM

10/2/2018 • 2 minutes to read • [Edit Online](#)

You can use Windows® System Image Manager (Windows SIM) to search for a component, setting, file in a distribution share, or package name by using the Find feature. The following procedure describes how to use Find.

1. Open Windows SIM.
2. On the **Edit** menu, click **Find**, or use the keyboard shortcut Ctrl+F. The **Find** dialog box appears.
3. In the **Find what** box, enter the search criteria.
4. In the **Look in** drop-down list, select from the currently open Windows image and answer file, a distribution share, or the **Messages** pane.
5. Click **Find Now**.

## Related topics

[Windows System Image Manager How-to Topics](#)

[Create or Open an Answer File](#)

[Configure Components and Settings in an Answer File](#)

[Validate an Answer File](#)

[Hide Sensitive Data in an Answer File](#)

[Add a Device Driver Path to an Answer File](#)

[Add a Package to an Answer File](#)

[Add a Custom Command to an Answer File](#)

# Create a Configuration Set

10/2/2018 • 2 minutes to read • [Edit Online](#)

The following procedure describes how to create a configuration set by using Windows® System Image Manager (Windows SIM).

A configuration set is a smaller version of a distribution share and is more easily copied to removable media or a network share. It is a collection of files that have been converted to binary form. These files are a self-contained alternative to referencing a distribution share.

Because a configuration set contains only internal references, it can be used for both online and offline installations. It can also be duplicated and changed for different types of installations.

1. Open Windows SIM.
2. Open an answer file. For more information, see [Create or Open an Answer File](#).
3. On the **Tools** menu, click **Create Configuration Set**. The **Create Configuration Set** window opens.
4. Browse to the destination folder for the configuration set, or enter a folder name.
5. Select a folder that you want to copy to your **\$OEM\$ Folders** folder (optional), and then click **OK**.

## IMPORTANT

If a configuration set is used during Windows Setup, all of the contents at the root of the media where the answer file exists are copied to the Windows installation. If there are many files and folders at the same level as the answer file, Windows Setup copies all of the files and folders to the Windows installation. Note that this might slow down installation. In some cases, you might run out of disk space.

## Related topics

[Windows System Image Manager How-to Topics](#)

[Manage Files and Folders in a Distribution Share](#)

[Add Packages to a Distribution Share](#)

[Create or Open a Distribution Share](#)

# Create or Open a Distribution Share

10/2/2018 • 2 minutes to read • [Edit Online](#)

A distribution share is an optional storage folder for third-party drivers, applications, and packages that Microsoft issues (such as updates).

You can create a distribution-share folder by using Windows® System Image Manager (Windows SIM) or by using a manual technique. The procedures in this topic describe how to create, open, and explore a distribution-share folder.

## NOTE

You must use Windows SIM to add packages. For more information, see [Add Packages to a Distribution Share](#).

## Create a distribution share using Windows SIM

1. Create a new folder where you want to place the distribution share. This folder can be on a network share (example: `\server\share\MyDistributionShare`) or on your local computer (example: `C:\MyDistributionShare`).
2. In the **Distribution Share** pane, right-click **Select a Distribution Share**, and then click **Create Distribution Share**. The **Create a Distribution Share** window appears.
3. Browse to the folder that you created, and then click **Open**. In the **Distribution Share** pane, the distribution-share folder opens.
4. Windows SIM automatically creates a folder structure for the distribution share.

## Create a distribution share manually

1. In Windows Explorer, create a new folder where you want to place the distribution share. This folder can be on a network share (example: `\server\share\MyDistributionShare`) or on your local computer (example: `C:\MyDistributionShare`).
2. In this folder, create the following subfolders:
  - **\$OEM\$ Folders**
  - **Packages**
  - **Out-of-Box Drivers**
  - **LangPacks**

## NOTE

Windows SIM recognizes only these subfolder names. For the distribution share to be valid, at least one of the four folders must be present. To enable Windows SIM to read the subfolder contents, the subfolder names must match this list exactly.

## Open a distribution share in Windows SIM

1. In the **Distribution Share** pane, click the top node of the currently open distribution share. Alternately, right-click **Select a Distribution Share**, and then click **Select Distribution Share**. The **Select a Distribution Share** dialog box opens.
2. Browse to the distribution share that you want to open. The distribution share can be opened only if the

following folder structure exists:

- **\$OEM\$ Folders**
- **Packages**
- **Out-of-Box Drivers**
- **LangPacks**

3. Select the distribution share that you want to open, and then click **OK**. The distribution share opens in the **Distribution Share** pane.

## Explore a distribution share from Windows SIM

1. In Windows SIM, right-click the top node in the **Distribution Share** pane, and then click **Explore Distribution Share**.
2. The distribution-share folder opens in Windows Explorer, where you can modify files or move files between folders.

## Related topics

[Windows System Image Manager How-to Topics](#)

[Manage Files and Folders in a Distribution Share](#)

[Add Packages to a Distribution Share](#)

# Manage Files and Folders in a Distribution Share

10/2/2018 • 2 minutes to read • [Edit Online](#)

After a distribution-share folder is created, you can add files to the **\$OEM\$ Folders** or **Out-of-Box Drivers** folders. You cannot add packages directly to the Packages folder. You must use Windows® System Image Manager (Windows SIM) to add packages to a distribution share. For more information, see [Add Packages to a Distribution Share](#).

You can make out-of-box device drivers (also called third-party drivers) available in Windows SIM by copying device drivers to the **Out-of-Box Drivers** folder in a distribution share. You can use subfolders to organize out-of-box drivers. When you add an **Out-of-Box Drivers** folder to an answer file, all drivers in the folder and subfolders are also added. After drivers are copied to the appropriate folder, they are available through Windows SIM and can be added to an answer file.

When you create a configuration set, you can use the **\$OEM\$ Folders** folder to copy scripts, binaries, and other files to Windows during installation. An answer file can reference files and folders stored in subfolders of **\\$OEM\$ Folders** can be referenced in an answer file. For more information, see [Create a Configuration Set](#).

## IMPORTANT

Do not overwrite existing files carried and serviced by the operating system. Overwriting system files can cause the operating system to behave unpredictably and cause serious issues.

To manage files and folders in a distribution share:

1. Open a distribution share. For more information, see [Create or Open a Distribution Share](#).
2. Right-click the distribution share, and then click **Explore Distribution Share**.
3. Double-click either the **\$OEM\$ Folders** folder or the **Out-of-Box Drivers** folder. The folder opens.
4. Manage files and folders in the following ways:
  - Create subfolders by right-clicking in the folder, clicking **New**, and then clicking **Folder**.
  - Add files to the distribution share by copying files and pasting them in the folder.
  - Delete distribution-share contents by right-clicking a file or folder and then clicking **Delete**.
  - Add out-of-box drivers by copying the device-driver files to the **Out-of-Box Drivers** folder.
  - Add applications, scripts, or other files to the **\$OEM\$ Folders** subfolders.
5. Close the distribution-share folder.
6. The changes appear in the **Distribution Share** pane.

## NOTE

The **\$OEM\$ Folders** subfolders are organized in a specific structure. Copy files to the **\$OEM\$ Folders** subfolders as described in [Distribution Shares and Configuration Sets Overview](#). For example, if you add files to **\$OEM\$ Folders\\$1\Program Files\Application1**, Windows Setup will copy them to **C:\Program Files\Application1** on the Windows installation.

## Related topics

[Windows System Image Manager How-to Topics](#)

[Create or Open a Distribution Share](#)

[Add Packages to a Distribution Share](#)

# Add Packages to a Distribution Share

10/2/2018 • 2 minutes to read • [Edit Online](#)

Packages are groups of files that Microsoft provides. These groups of files include service packs, security updates, language packs, and modifications to Windows® features. You can add a package to a Windows installation by using an answer file, a configuration set, or a distribution share.

You must use Windows System Image Manager (Windows SIM) to import packages. After a package is imported and available in a distribution share, you can add the package to an answer file. For more information, see [Add a Package to an Answer File](#).

## NOTE

For specific information about how to add language packs, see [Add Multilingual Support to a Windows Distribution](#).

To import a package to a distribution share:

1. Select and open a distribution share. For more information, see [Create or Open a Distribution Share](#).
2. On the **Tools** menu, click **Import Package(s)**. The **Select Package(s) to Import** window opens.
3. Browse to the file or folder, select the file or folder, and then click **Open** or **Open Folder**.

Windows SIM adds the selected package to the distribution-share folder. The newly added package is displayed under the **Packages** node in the **Distribution Share** pane.

## Related topics

[Windows System Image Manager How-to Topics](#)

[Manage Files and Folders in a Distribution Share](#)

# Windows System Image Manager Reference Topics

10/2/2018 • 2 minutes to read • [Edit Online](#)

The following topics provide reference information about Windows System Image Manager (Windows SIM).

## In This Section

<a href="#">Component Settings and Properties Reference</a>	Describes the structure of answer files, along with the attributes and elements that components and settings use.
<a href="#">Windows System Image Manager Architecture</a>	Describes how Windows SIM works.
<a href="#">Windows System Image Manager Supported Platforms</a>	Lists the supported platforms where you can install Windows SIM.

## Related topics

[Windows Deployment Options](#)

[Windows System Image Manager Technical Reference](#)

# Component Settings and Properties Reference

10/2/2018 • 7 minutes to read • [Edit Online](#)

Windows System Image Manager (Windows SIM) displays the properties and settings of a selected component or package in the **Properties** pane. You can use this pane to manage and view the component settings that are available to change for each configuration pass. You can also use this pane to view properties and IDs where applicable. In the case of packages, the pane displays Windows feature selections that you can change. Settings that are not available for each component or package appear dimmed.

## Component Settings

Component settings are the configurable aspects of each component in a Windows installation. For example, you can configure the Windows Internet Explorer component setting **Home\_Page** to open to a particular URL by configuring the default value of the setting in the **Properties** pane of Windows SIM.

## Component Properties

Component properties are non-configurable attributes of a component. The following table lists component properties for components that have been added to an answer file.

PROPERTY	DESCRIPTION
<b>AppliedConfigurationPass</b>	Specifies the configuration pass that all child settings are applied to.
<b>Component</b>	Specifies the root <b>ComponentSetting</b> object that this setting override belongs to.
<b>Path</b>	Specifies the path to the setting from the component. The path appears in the following format: <i>SettingName1/SettingName2/...</i>
<b>Enabled</b>	Indicates whether the component has been installed. A setting of <b>True</b> means that the component is installed. A setting of <b>False</b> means that the component is not installed. When the component is not installed, the setting is ignored and the correct Windows Features in the foundation package that contains the component are enabled.

## Component IDs

The component ID uniquely identifies the component of the operating system to which the settings belong. The ID contains the name, version, architecture, and other information for the component that is selected in the **Windows Image** pane or **Answer File** pane. The following table describes the different attributes of a component.

ID	DESCRIPTION
<b>Language</b>	Specifies the language code. For more information, see the language codes in the <a href="#">MSDN Library</a> .
<b>Name</b>	Specifies the long name of the component or package.
<b>ProcessorArchitecture</b>	Specifies the processor architecture of the component or package. For example, <b>x86</b> or <b>amd64</b> .
<b>PublicKeyToken</b>	Specifies the public-key token of the component or package. This is a string of 16 hexadecimal digits and is the hash value of the Microsoft public key. The value is unique and prevents collision between components and packages.
<b>Version</b>	Specifies the version of the Windows component or package.
<b>VersionScope</b>	Specifies the version scope of the Windows component or package. The possible values are <b>SxS</b> and <b>nonSxS</b> .

## Package Properties

Package properties are non-configurable attributes of the package. Package properties appear when you select a package in the **Windows Image** pane or **Answer File** pane. The following table describes the properties of packages.

PACKAGE PROPERTY	DESCRIPTION
<b>CompanyName</b>	Specifies the name of the company that created the package.
<b>Copyright</b>	Specifies the copyright disclaimer of the package.
<b>Description</b>	Specifies the description of the package.
<b>Id</b>	Specifies the identifier for the package. The format is: <i>ProcessorArchitectureVersionLanguagePublicKeyTokenVersionScope</i>
<b>Keyword</b>	Specifies the keyword of the package.
<b>Path</b>	Specifies the file-system path of the package file. This is blank if the package is from a Windows image.

PACKAGE PROPERTY	DESCRIPTION
<b>ProductName</b>	Specifies the product name that this package applies to.
<b>ProductVersion</b>	Specifies the product version that this package applies to.
<b>ReleaseType</b>	Specifies the <b>PackageReleaseType</b> enumeration of this package. <b>PackageReleaseType</b> is documented in the Component Platform Interface (CPI) Reference.
<b>SupportInformation</b>	Specifies the support information for the package. This can contain contact information about the package author.

## Package Settings

Package settings are the configurable attributes of the package that is selected in the **Answer File** pane. Package settings appear only when the package is selected in the **Answer File** pane because that is when you can change them. The following table describes package settings

SETTING NAME	DESCRIPTION
<b>Action</b>	Specifies the action to be performed on the package within the answer file. Possible actions are <b>Install</b> , <b>Configure</b> , <b>Remove</b> , or <b>Stage</b> .
<b>PermanenceType</b>	Describes whether a component is removable or permanent. Permanence types are members of the <b>PackageActionType</b> enumeration and are documented in the CPI Reference (CPIAPI.chm).
<b>PrimarySourcePath</b>	Specifies the primary file-system path that is the source of the package file. If the package is from a Windows image, this will be blank.

## Right-Click Menu Options

The following menu commands are available when you right-click a setting in the **Properties** pane.

COMMAND	DESCRIPTION
<b>revert change</b>	Reverts to the previous state or setting. This command removes the entry for the setting from the answer file. The setting remains unchanged after the Unattend.xml answer file has been applied.

COMMAND	DESCRIPTION
<b>write empty string</b>	<p>Writes the XML equivalent of an empty string for the setting in the answer file.</p> <p>By default, if no value is specified, the setting will be omitted from the answer file. However, you can specifically write an empty value for a string type in an answer file by using this command.</p> <p>This command applies to string types only.</p> <div style="border: 1px solid black; padding: 5px;"> <p><b>Important</b></p> <p>Not all component string settings support empty values. For more information, see the Unattended Windows Setup Reference.</p> </div>
<b>write image value</b>	Creates an entry for the setting in the answer file with the value of the setting that is currently in the Windows image.

## .NET Types in Windows System Image Manager

Microsoft® .NET types appear at the bottom of the **Properties** pane. Component settings have a type that describes the kind of data that is valid for that setting. These types are mapped to their equivalent .NET types in Windows SIM. The following table lists the possible types that can be associated with component settings.

.NET TYPE	PARAMETERS	DESCRIPTION
<b>System.Byte</b>	0 to 255	Unsigned 8-bit integer
<b>System.SByte</b>	-128 to 127	Signed 8-bit integer
<b>System.UInt16</b>	0 to 65,535	Unsigned 16-bit integer
<b>System.Int16</b>	-32,768 to 32,767	Signed 16-bit integer
<b>System.UInt32</b>	0 to 4,294,967,295	Unsigned 32-bit integer
<b>System.Int32</b>	-2,147,483,648 to 2,147,483,647	Signed 32-bit integer
<b>System.UInt64</b>	0 to 18,446,744,073,709,551,615	Unsigned 64-bit integer
<b>System.Int64</b>	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	Signed 64-bit integer
<b>System.Boolean</b>	true   false	Boolean data

.NET TYPE	PARAMETERS	DESCRIPTION
<b>System.String</b>	Represents text as a series of Unicode characters	String data

## Array Types

Some component settings require arrays of data. These arrays are mapped to their equivalent .NET array types in Windows SIM. The following table lists the possible array types that are associated with component settings.

TYPE	DESCRIPTION
<b>System.String[]</b>	Array of <b>System.String</b>
<b>System.Byte[]</b>	Array of <b>System.Byte</b>
<b>System.SByte[]</b>	Array of <b>System.SByte</b>

## List-Item Types

Settings are sometimes organized into groups called list items. List items specify one or more values for a list-item type. A list-item type may include one or more component settings. For example, you can create multiple Favorites links by using the **FavoriteItem** setting for Internet Explorer.

You add a list item by right-clicking the setting for a container. For example, you can add a **FavoriteItem** list item by right-clicking the **FavoritesList** container in the Answer File pane. For more information, see [Configure Components and Settings in an Answer File](#).

### Key Settings for List Items

Each list item must have a unique identifier, which is known as the key for that specific list item. When you modify the key setting for the list item, the key identifier appears in brackets ([]) next to the list item in the **Answer File** pane.

### List-Item Actions

The following actions are available for list items when you use Windows SIM.

#### Add a New List Item

You can use Windows SIM to add list items to the currently open answer file. In the **Setting Action** drop-down list, click **AddListItem**. You must also add a unique key setting to the list item. The unique key setting appears in brackets next to the list item in the tree view of the **Answer File** pane. A plus sign (+) appears, which indicates that the list item is added to the Windows image when the unattended answer file is run.

#### Delete a List Item

You can use Windows SIM to delete a list item that is defined in a Windows image (.wim) file. In the **Setting Action** drop-down list, click **RemoveListItem**. A minus sign (-) appears, which indicates that the list item is deleted from the image when the unattended answer file is run.

#### Modify a List Item

You can use Windows SIM to modify a list item that is defined in a Windows image file. To change the default value for an existing list item, click **Modify** in the **Properties** pane, and then enter the updated information under **Settings**. The updated list-item setting is added to the answer file.

## Related topics

[Windows System Image Manager Reference Topics](#)

[Windows System Image Manager Overview Topics](#)

# Windows System Image Manager Architecture

10/2/2018 • 2 minutes to read • [Edit Online](#)

You use Windows® System Image Manager (Windows SIM) to create an XML-based answer file that is required to automate Windows installations.

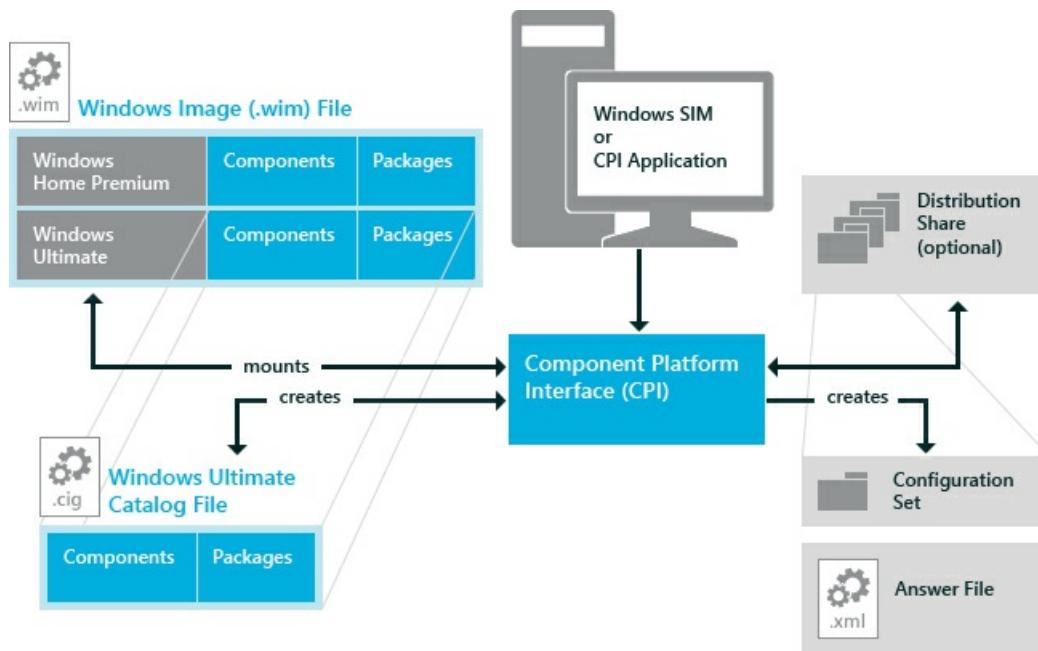
Windows SIM uses the Component Platform Interface (CPI API) to create and manage answer files. The components and settings in a specific Windows image are used to create a catalog file. This catalog file is used in Windows SIM to create answer files. For more information, see [Windows Image Files and Catalog Files Overview](#).

An optional set of folders, called a distribution share, can be created to store files that you use to further customize your Windows installation. For more information, see [Distribution Shares and Configuration Sets Overview](#).

Windows SIM can create a smaller, more portable version of a distribution share called a configuration set. These smaller files can be easier to manage.

You can also use the CPI API to create your own customized applications that can automate the creation and management of unattended Windows Setup answer files. For more information, see the [Component Platform Interface \(CPI\) Reference \(CPIAPI.chm\)](#).

The following diagram shows how Windows SIM works.



## Related topics

[Windows System Image Manager Reference Topics](#)

[Windows System Image Manager Overview Topics](#)

# Windows System Image Manager Supported Platforms

10/2/2018 • 2 minutes to read • [Edit Online](#)

## Supported Platforms

By using the 32-bit version of Windows System Image Manager (Windows SIM), you can create and open catalog (.clg) files for Windows images of all architecture types. You can use the 64-bit versions of Windows SIM to create catalog files for only 64-bit Windows images. For example, you can use the 64-bit version of Windows SIM to create catalog files for only 64-bit-based Windows images.

After you create a catalog file, you can open catalog files for all Windows image architecture types.

Windows Preinstallation Environment (Windows PE) is not a supported platform for Windows SIM.

The following table lists Windows operating systems and the supported list of architecture types on which you can create catalog files for Windows images.

OPERATING SYSTEM	SUPPORTED ARCHITECTURE FOR .CLG CREATION
Windows 10, build 1803 (64-bit edition)	x64-based
Windows 10, build 1709 (64-bit edition)	x64-based
Windows 10, build 1703 (64-bit edition)	x64-based
Windows 10 Server	x64-based
Windows 10, build 1607 (64-bit edition)	x64-based
Windows 10, build 1511 (64-bit edition)	x64-based
Windows 10 (64-bit edition)	x64-based
Windows Server 2012 R2 (64-bit edition)	x64-based
Windows Server 2012 (64-bit edition)	x64-based
Windows 8.1 (64-bit edition)	x64-based
Windows 8.1 (32-bit edition)	x86-based, x64-based, Windows RT ARM-based

OPERATING SYSTEM	SUPPORTED ARCHITECTURE FOR .CLG CREATION
Windows 8 (32-bit edition)	x86-based, x64-based, Windows RT ARM-based
Windows 8 (64-bit edition)	x64-based
Windows Server 2008 R2 (64-bit edition)	x64-based only
Windows Server 2008 R2 (Itanium-based)	Itanium-based only
Windows Server 2008 (Itanium-based)	Itanium-based only
Windows Server 2008 (64-bit edition)	x64-based only
Windows Server 2008 (32-bit edition)	x86-based, x64-based, Itanium-based
Windows Server 2003 with SP2 (64-bit edition)	x64-based only
Windows Server 2003 with SP2 (32-bit edition)	x86-based, x64-based, Itanium-based
Windows 7 (64-bit edition)	x64-based only
Windows 7 (32-bit edition)	x86-based, x64-based, Itanium-based
Windows Vista with SP1 and SP2 (64-bit edition)	x64-based only
Windows Vista with Service Pack 1 (SP1) and Service Pack 2 (SP2) (32-bit edition)	x86-based, x64-based, Itanium-based

## Related topics

[Windows System Image Manager Reference Topics](#)

# Unattended Windows Setup Reference

10/2/2018 • 2 minutes to read • [Edit Online](#)

The Windows Unattended Setup Reference provides a complete listing of all the settings that you can use to automate the configuration and the deployment of Windows 10.

The Windows Unattended Setup Reference is organized by Windows components and Windows packages, in the same order that the Windows System Image Manager (Windows SIM) tool displays each Windows component and package.

Each Windows component includes settings that can be used to create an unattended-installation answer file. Each setting in a component is listed in its own individual topic. If an element contains a value, valid value types are described and XML examples are given.

Information about how to use Windows SIM is available in the [Windows System Image Manager Technical Reference](#).

## NOTE

All Unattend settings for Windows 10 are also supported in S mode, with the exception of [Microsoft-Windows-Shell-Setup-FirstLogonCommands](#).

## In this section

TOPIC	DESCRIPTION
<a href="#">Changed answer file settings for Windows 10 for desktop editions, version "RS5"</a>	This topic describes Windows 10, version "RS5" answer-file settings that have changed since Windows 10 for desktop editions, version 1803.
<a href="#">Changed answer file settings for Windows 10 for desktop editions, version 1803</a>	This topic describes Windows 10, version 1803 answer-file settings that have changed since Windows 10 for desktop editions, version 1709.
<a href="#">Changed answer file settings for previous Windows 10 builds</a>	This topic shows the historic list of changes to answer file settings for each build of Windows 10 that was released prior to build 1803.
<a href="#">Components</a>	The topics in this section describe all of the unattended settings that can be set in Windows 10 and Windows Server 2016. To determine whether a component applies to the image you're building, load your image into Windows SIM and search for the component or setting name. For information on how to view components and settings, see <a href="#">Configure Components and Settings in an Answer File</a>

# Customizations for mobile devices

10/2/2018 • 2 minutes to read • [Edit Online](#)

Customizations for Windows 10 Mobile allow you to run mobile line-of-business applications on a platform that ensures that data is captured securely and efficiently.

## TIP

Before getting started with customizations, review the prerequisite requirements, and download the tools needed to customize, test, and deploy Windows on mobile devices. See [Prepare for Windows mobile development](#) for guidance.

## In this section

TOPIC	DESCRIPTION
<a href="#">Enterprise shared storage</a>	Enterprise shared storage defines local data locations for line of business apps to share data.
<a href="#">Customize using the mobile MCSF</a>	The Managed Centralized Settings Framework (MCSF) is part of the customization and multivariant infrastructure first introduced in Windows Phone 8.1 and is still supported in Windows 10 Mobile. MCSF consists of both image time and runtime components that enable these functionality. MCSF creates configuration service providers for registry-backed settings and custom configuration service providers can be referenced for more complex settings.

## Related topics

[Prepare for Windows mobile development](#)

[Create mobile packages](#)

# Enterprise shared storage

10/2/2018 • 2 minutes to read • [Edit Online](#)

Enterprise shared storage defines local data locations for line of business apps to share data.

The shared storage consists of two locations, where apps with the restricted capability **enterpriseDeviceLockdown** and an Enterprise certificate have full read and write access. Note that the **enterpriseDeviceLockdown** capability allows apps to use the device lock down API and access the enterprise shared storage folders. For more information about the API, see [Windows.Embedded.DeviceLockdown](#) namespace.

These locations are set on the local drive:

- \Data\SharedData\Enterprise\Persistent
- \Data\SharedData\Enterprise\Non-Persistent

## Scenarios

Enterprise shared storage provides support for the following scenarios.

- You can share data within an instance of an app, between instances of the same app, or even between apps assuming they both have the appropriate capability and certificate.
- You can store data on the local hard drive in the \Data\SharedData\Enterprise\Persistent folder and it persists even after the device has been reset.
- Manipulate files, including read, write, and delete of files on a device via Mobile Device Management (MDM) service. For more information on how to use enterprise shared storage through the MDM service, see [EnterpriseExtFileSystem CSP](#).

## Access enterprise shared storage

The following example shows how to declare the capability to access enterprise shared storage in the package manifest, and how to access the shared storage folders by using the Windows.Storage.StorageFolder class.

In your app package manifest, include the following capability:

```
<Package
  xmlns="http://schemas.microsoft.com/appx/manifest/foundation/windows10"
  xmlns:mp="http://schemas.microsoft.com/appx/2014/phone/manifest"
  xmlns:uap="http://schemas.microsoft.com/appx/manifest/uap/windows10"
  xmlns:rescap="http://schemas.microsoft.com/appx/manifest/foundation/windows10/restrictedcapabilities"
  IgnorableNamespaces="uap mp rescap">

  ...
  <Capabilities>
    <rescap:Capability Name="enterpriseDeviceLockdown"/>
  </Capabilities>

```

To access the shared data location, your app would use the following code.

```
using System;
using System.Collections.Generic;
using System.Diagnostics;
using Windows.Storage;

...
// Get the Enterprise Shared Storage folder.
var enterprisePersistentFolderRoot = @"C:\Data\SharedData\Enterprise\Persistent";

StorageFolder folder =
    await StorageFolder.GetFolderFromPathAsync(enterprisePersistentFolderRoot);

// Get the files in the folder.
IReadOnlyList<StorageFile> sortedItems =
    await folder.GetFilesAsync();

// Iterate over the results and print the list of files
// to the Visual Studio Output window.
foreach (StorageFile file in sortedItems)
    Debug.WriteLine(file.Name + ", " + file.DateCreated);
```

# Customize using the mobile MCSF framework

10/2/2018 • 3 minutes to read • [Edit Online](#)

The Managed Centralized Settings Framework (MCSF), which was introduced in Windows Phone 8.1, is supported in Windows 10.

MCSF consists of both image time and runtime components that enable a simple and consistent way of declaring OS settings, and a centralized framework that the runtime configuration engine, mobile operators, device management servers, and others can query or modify these settings. A runtime configurable MCSF can either be a configuration service provider-based customization or registry-based customization being exposed through the MSCF CSP.

Similar to the Unattend answer file that can be provided to Windows Setup for Windows Desktop image customization, a customization answer file can be passed to customize an image with specific settings and to create variants for the image. The customization answer file allows for a broader integration across the OS by providing OEMs with a single place to define nearly all OS settings. During image build time, the customization answer file is processed and built as customization packages, which are automatically included in the images. OEM partners can leverage this system by defining a registry or configuration service provider-based settings in packages using MCSF. MCSF also produces customization policy files for both OEM partners and Microsoft. These files are extracted from the packages used to build the OS image to determine the valid settings for the OS image.

## In this section

TOPIC	DESCRIPTION
<a href="#">Managed Centralized Settings Framework (MCSF)</a>	Provides a standard way to describe settings that are customizable within packages. MCSF also generates a policy based on the settings descriptions. The settings framework can be image time or runtime configurable. A runtime configurable MCSF can either be a configuration service provider-based customization or registry-based customization being exposed through the MSCF CSP.
<a href="#">Customization answer file</a>	A <b>customization answer file</b> is an XML file that you write based on the MCSF schema. OEMs can use the MCSF customization answer file to specify the settings and variants for a custom mobile OS image. The customization answer file allows for a broader integration across the OS by providing OEMs with a single place to define nearly all mobile OS settings.
<a href="#">Set languages and locales</a>	Provides an overview of the different language and locale settings based on the market in which the mobile device will ship.
<a href="#">Create a resource-only .dll for localized strings</a>	
<a href="#">Customizations for device management</a>	This section provides more information about device management settings that OEMs can change.

TOPIC	DESCRIPTION
Customizations for hardware components	<p>This section contains information about customization settings that OEMs can use for the following hardware components:</p> <ul style="list-style-type: none"> <li>• Buttons</li> <li>• Camera</li> <li>• Display</li> <li>• Networking</li> <li>• Sensors</li> <li>• Storage</li> <li>• Touch</li> </ul>
Customizations for applications and Microsoft components	This section contains information about customizations related to apps and Microsoft components.
Customizations for boot, initial setup, and shutdown	Use these customizations to configure the device boot, initial setup, or shutdown experience.
Customizations for browser	Describes the customizations for the browser.
Customizations for connectivity	Describes the customizations for configuring connectivity settings.
Customizations for desktop experiences	Describes the customizations that you can configure for the desktop when the mobile device is connected.
Customizations for email	Describes the customizations related to email.
Customizations for maps	Describes the customizations that you can configure for maps on the mobile device.
Customizations for phone calls	Provides information about customizations you can configure for the phone or dialer app including branding, visual voicemail, caller ID matching, dialer codes, and more.
Customizations for photos, music, and videos	Contains the customizations you can configure for photos, music, and videos.
Customizations for Settings	The <b>Settings</b> app contains a predefined collection of user-configurable system settings that's organized into pages by functionality. As specified in policy, the appearance and default values of these settings are generally not customizable. The following table contain the complete list of user-facing settings that OEMs and mobile operators can change.

TOPIC	DESCRIPTION
<a href="#">Customizations for SMS and MMS</a>	Contains settings that you can configure for SMS and MMS.
<a href="#">Customizations for Start</a>	This section contains information about customizations related to Start.

## Related topics

[Prepare for Windows mobile development](#)

# Managed Centralized Settings Framework (MCSF)

10/2/2018 • 11 minutes to read • [Edit Online](#)

The Managed Centralized Settings Framework (MCSF) is part of the customization and multivariant infrastructure first introduced in Windows Phone 8.1 and is still supported in Windows 10 Mobile. This component provides:

- A simple and consistent way for Microsoft to declare various mobile OS settings.
- All the necessary features and settings needed for mobile OS image and multivariant customization to be declared and fully supported.
- A centralized framework that the provisioning engine, mobile operators, device management servers, and others can query or modify these settings.

MCSF consists of both image time and runtime components that enable these functionality. MCSF creates configuration service providers for registry-backed settings and custom configuration service providers can be referenced for more complex settings.

OEM partners can use MCSF and the MCSF packaging XML schema to declare and access custom OEM settings. The following sections provide more information about how you can declare your custom settings that conform to the MCSF packaging XML schema.

## Mobile OS settings and customizations

Microsoft-owned policy settings are documented in the `Microsoft.FeatureArea.FeatureSubArea.policy.xml` files. If you have the Adaptation Kit installed, you can find the policy settings files in the `%WPKCONTENTROOT%\OEMCustomization\generatedPolicy` directory.

### Note

The packaging and imaging tools do not read the policy settings from these XML files. The tools read the policy settings from internal policy configurations that are not available for partners to modify directly. If you modify the XML files in the `generatedPolicy` directory, nothing will happen.

These XML files are provided only for your convenience so that you can review the customizations that have been enabled on Windows 10 Mobile.

## Declaring settings

Declared settings are added in the components' existing `.pkg.xml` file as children of any **Component** element. For example, the following XML exposes a simple **DWORD** in the registry as a setting in a package file:

```
<SettingsGroup Path="OSArea/Feature">
    <Setting Name="Setting" Description="This is a DWORD registry value.">
        <RegistrySource Type="REG_DWORD" Path="HKEY_LOCAL_MACHINE\Software\Sample\RegKey" />
    </Setting>
</SettingsGroup>
```

### Settings groups

Settings are grouped within the package file. A **SettingsGroup** element represents a settings group in the [customization answer file](#). This is the top-level element for the MCSF packaging XML. Each group can contain any number of individual settings.

OEMs building their own flexible hierarchy of settings must choose an associated **Path** for each group of settings. For example, ringtone settings may reside in a "Shell\Ringtones" group while lock screen settings may reside under a "Shell\Wallpaper" group. Microsoft recommends using a **Path** naming scheme similar to the above example: "OSArea/Feature"

### Note

When naming your settings group, you must not include illegal file system characters in the settings group name. This [MSDN Web site](#) provides some guidelines on naming conventions. You can use both \ and / but do not use the reserved characters in your settings group name.

### Group constraints

Some settings groups may have special properties. A **Constraints** element within the settings group declaration indicates these special properties.

ATTRIBUTE	DESCRIPTION	EXAMPLE
<b>ImageTimeOnly</b>	<p>Specifies settings that are available to OEMs for customization during image time, but does not require runtime configuration by the runtime configuration engine or OTA.</p> <p><b>Note</b> If you are declaring a settings group that is not in the MainOS partition, you must specify this settings group as image time only.</p>	<pre>&lt;Constraints ImageTimeOnly="Yes" /&gt;</pre>
<b>FirstVariationOnly</b>	<p>Specifies settings that are restricted such that these settings can only be modified by the runtime configuration engine during the first variation (typically when applying branding related to the first SIM).</p> <p>Settings with the <b>FirstVariationOnly</b> constraint are configured whenever the runtime configuration engine finds the first valid configuration when a SIM is inserted and there is a marked configuration for it. During SIM change, the value for the <b>FirstVariationOnly</b> setting will not be changed again.</p>	<pre>&lt;Constraints FirstVariationOnly="Yes" /&gt;</pre>
<b>Atomic</b>	<p>Specifies that one or more of the settings depend on the value of some of the other settings in the group. For OEMs that configures an atomic settings group, OEMs must specify every setting in that group.</p>	<pre>&lt;Constraints Atomic="Yes" /&gt;</pre>

## Individual settings

A **Setting** element represents a setting that is contained within a **Settings** group. Each setting must have a **Name** attribute, which must be unique within the settings group. A **Description** attribute can be used to display information about the setting. For example, a description can be: "Use to display the battery life warning on phones with AMOLED or OLED displays." OEMs must provide a **Description** attribute for each setting.

A setting can point to a particular location in the registry or to a configuration service provider node.

- To point your setting at a registry location, add a **RegistrySource** element to the setting.

The **RegistrySource** must contain a **Path** attribute that specifies the full path to the desired value in the registry. You must also specify the **Type** attribute, such as REG\_DWORD, REG\_SZ, and so on, to indicate the type of registry value. Optionally, you can specify a default value for your registry setting by including a **Default** value. The following example shows what a registry-sourced setting looks like:

```
<Setting Name="MyString">
    <!-- Use '@' to specify the default registry value. The Path must resolve to a value, rather than a key.
        Note that this registry location will have a default value of 3. -->
    <RegistrySource Type="REG_SZ" Path="HKEY_LOCAL_MACHINE\Software\Sample\@" Default="DefaultValue" />
</Setting>
```

- To point your setting at a configuration service provider node, add a **CSPSource** element to the setting.

The **CSPSource** must contain a **Path** attribute that specifies the full URI path to the desired configuration service provider node. You must also specify the **Type** attribute, which may contain one of the ConfigManager2 data types such as CFG\_DATATYPE\_INTEGER, CFG\_DATATYPE\_STRING, and so on. The following example shows what a configuration service provider-sourced setting looks like:

```
<Setting Name="MyPhoneVersion">
    <CspSource Type="CFG_DATATYPE_STRING" Path=".//devdetail/swv" />
</Setting>
```

## Controlling access

You can control access to each setting by including an optional **AccessType** element. There are four **AccessType** attributes: **Create**, **Delete**, **Get**, and **Replace**. There is also an **All** attribute that overrides any of the other access types. These attributes default to "Yes" so if you omit **AccessType** altogether, this implies that all access is allowed for your setting. You can use this element if you need to restrict the allowed operations for your setting.

The following example shows how to specify the type of access for a setting:

```
<Setting Name="NoDeleteSetting">
    <RegistrySource Type="REG_DWORD" Path="HKEY_LOCAL_MACHINE\Software\Sample\NoDelete" Default="3" />

    <!-- This Setting cannot be deleted once created. -->
    <AccessType Create="Yes" Delete="No" Get="Yes" Replace="Yes" />
</Setting>
```

## Note

This access control is done at the MCSF configuration service provider level and is not intended to take the place of security. If you need to protect your registry locations, you must do so using capabilities.

## Validating settings

MCSF allows you to specify simple rules to control which values are allowed for your settings. To do this, add a **Validate** element for your setting. There are three methods to validate a setting:

- For numeric values, you can specify a minimum and/or a maximum value. You can do this by setting the **Min** and **Max** attributes in the **Validate** element.

```
<Setting Name="MyNumericSetting">
  <RegistrySource Type="REG_DWORD" Path="HKEY_LOCAL_MACHINE\Software\Sample\Number" Default="3" />

  <!-- Validate that the range is from 0 through 10. -->
  <Validate Min="0" Max="10" />
</Setting>
```

- For string values, you can specify a minimum and/or a maximum string length. You can do this by setting the **MinLength** and **MaxLength** attributes in the **Validate** element.

```
<Setting Name="MyString">
  <RegistrySource Type="REG_SZ" Path="HKEY_LOCAL_MACHINE\Software\Sample\PutStringHere" Default="" />

  <!-- Validate that the string length is from 0 to 255. -->
  <Validate MinLength="0" MaxLength="255" />
</Setting>
```

- You can also specify an explicit list of allowed values for your setting. You can do this by adding the **Option** element to the **Validate** element for each allowed value. Option elements have a **Value** attribute and an optional **FriendlyName** attribute. You can use the friendly name to show identifiable string names in your customization tools for simple enum values. The following example shows several options with friendly names:

```
<Setting Name="MyEnum">
  <RegistrySource Type="REG_DWORD" Path="HKEY_LOCAL_MACHINE\Software\Sample\PutOneOfListHere"
Default="1" />

  <!-- The Value attribute has to be one of the following values. -->
  <Validate>
    <Option Value="0" FriendlyName="Red" />
    <Option Value="1" FriendlyName="Green" />
    <Option Value="2" FriendlyName="Blue" />
  </Validate>
</Setting>
```

Options for numeric settings must be specified in hexadecimal format, but without a 0x prefix.

### Settings that reference files

Some settings may be related to asset files, such as lock screen backgrounds, that OEMs or mobile operators include in the phone image. To specify an asset file, use an **Asset** element to your settings group. Each **Asset** element must specify a unique asset **Name** attribute as well as a **Description**. Assets are defined by a number of properties that help the imaging system validate their content.

The asset contains a **Type** attribute that specifies a semicolon delimited list of file extensions that the setting supports. OEMs must also specify a **Path** attribute to indicate where the OEM assets should be placed on the phone.

Any **Setting** can reference any existing asset by specifying the **Asset** attribute. This indicates to the OEM customization tools that the setting expects to use an asset filename for its value.

### Note

When creating your own policy settings and you're adding an **Asset** to a **SettingsGroup**, you must specify the **Asset** within the **SettingsGroup** first. Otherwise, the package containing your policy settings will not get built.

The following example shows how an **Asset** as the first attribute declared within the **SettingsGroup** and a **Setting** that references the **Asset** that was declared:

```
<SettingsGroup Path="LockScreen">
    <Asset Name="Wallpapers" Type=".jpg;.jpeg;.png" Path="\Programs\CommonFiles\Wallpapers" Description="Use to add lock screen backgrounds to the phone." >
        <MultiStringList Key="HKEY_LOCAL_MACHINE\Software\Microsoft\Shell\OEM\Wallpaper" Value="WallpaperSet" />
    </Asset>

    <Setting Name="DefaultWallpaper" Asset="Wallpapers" Description="Use to set the default lock screen background.">
        <RegistrySource Type="REG_SZ" Path="HKEY_LOCAL_MACHINE\Software\Microsoft\Shell\OEM\Wallpaper\CurrentWallpaper" />
    </Setting>
</SettingsGroup>
```

**Asset** names can include macros and allow you to put files into locations based on the value of the macro specified in the asset name.

### Warning

Macros in **SettingsGroup** paths do not work with Assets. To use a macro in an **Asset** path, the macro reference must be part of the **Asset** name.

```
<SettingsGroup Path="Contoso/MyConfig">
    <Asset Name="MyVariant/${(VarType)}"
          Type=".txt"
          Path="\Programs\CommonFiles\OEM\Public\Contoso\${(VarType)}\" Description="Use to write a text file" />
</SettingsGroup>
```

Optionally, the image customization process can build a list of supplemental asset files added by OEM partners or mobile operator partners and set the list in the registry. There are two models that support this:

- OEMs can use the **ValueList** element within your asset declaration if your component expects a list of extra files to appear in the registry as individual values per file under a parent key. The **ValueList** element has two attributes, **OEMKey** and **MOKey**, which specify where OEM and mobile operator files should be listed, respectively. The customization tooling creates one value for each asset file added by the OEM or mobile operator. These values will be of registry type REG\_SZ. The name will be the full path to the asset file on the device. The REG\_SZ content of the value will be a localizable string in the form "`<MUID11Path>,-<StringID>`". The **ValueList** can also specify the **FileNamesOnly** attribute to indicate that it does not expect full paths in the list, only file names.

The following example shows this model:

```
<Asset Name="Ringtones" Type=".wma" Path="\Programs\CommonFiles\Sounds">
    <ValueList OEMKey="HKEY_LOCAL_MACHINE\Software\Microsoft\Shell\OEM\Sounds\Ringtones"
               MOKey="HKEY_LOCAL_MACHINE\Software\Microsoft\Shell\MO\Sounds\Ringtones" FileNamesOnly="Yes" />
</Asset>
```

- OEMs can use the **MultiStringList** element in your asset if your component expects a simple REG\_MULTI\_SZ list of file paths in the registry. The customization tooling builds a multi-string list of each asset file added by partners and sets this value at the registry location specified by the **Key** and **Value** attributes.

### Note

In Windows 10 Mobile, the Windows Provisioning framework does not support multi-string registry values so this element is only available in the MCSF framework. This means that assets, such as wallpapers or ringtones, can only be added through MCSF.

The following example shows this model:

```
```XML
<Asset Name="Wallpapers" Type=".jpg;.jpeg;.png" Path="\Programs\CommonFiles\Wallpapers">
    <MultiStringList
        Key="HKEY_LOCAL_MACHINE\Software\Microsoft\Shell\OEM\Wallpaper"
        Value="WallpaperSet" />
</Asset>
...```

```

## Multi-settings

Multi-settings are settings or groups of settings that are SIM-based or account-based. You can handle these types of settings using MCSF by declaring your **SettingsGroup's Path** in a specific format that declares one segment of the setting's URI to become a variable. You can reuse this variable when you declare each of your settings' underlying source locations such as the **RegistrySource path** or **CSPSource Path**.

To declare a multi-setting variable, you can use the **\$(VariableName)** packaging XML macro format. The following examples show how to do this:

```
<!-- Specifies a pair of Settings that we may have many sets of, one per "account" -->
<SettingsGroup Path="TestSettings/MyAccounts/${AccountId}">

    <!-- Each account has a username. -->
    <Setting Name="Username">
        <!-- Note that the macro used above in the SettingsGroup must also appear here. -->
        <RegistrySource Type="REG_SZ" Path="HKEY_LOCAL_MACHINE\Software\Sample\${AccountId}\Username" />
    </Setting>

    <!-- Each account has a password. -->
    <Setting Name="Password">
        <RegistrySource Type="REG_SZ" Path="HKEY_LOCAL_MACHINE\Software\Sample\${AccountId}\Password" />
    </Setting>

</SettingsGroup>
```

In the above example, the variable name is **AccountId**. You must specify that segment of the setting's URI to provision this setting. In this example, when MCSF sets the underlying value, it fetches the string value of the **AccountId** segment and injects it into the registry paths for Username and Password. For example, provisioning the following XML sets `HKEY_LOCAL_MACHINE\Software\Sample\{71986}\Username` to "RobinNail" and `HKEY_LOCAL_MACHINE\Software\Sample\{71986}\Password` to "1234Password".

```
<wap-provisioningdoc>
    <characteristic type="MCSF">
        <!-- This sets the username and password for account {71986}. -->
        <characteristic type="TestSettings\MyAccounts\{71986}">
            <parm name="Username" value="RobinNail" />
            <parm name="Password" value="1234Password" />
        </characteristic>
    </characteristic>
</wap-provisioningdoc>
```

For SIM-based settings, you can use the special variable **\$(\_\_IMSI)** that's built-in to MCSF. When these types of settings are being provisioned, the **\$(\_\_IMSI)** segment is replaced with the IMSI string for the current SIM card. The following example shows how to do this:

```
<!-- Specifies some SIM-specific settings... -->
<SettingsGroup Path="TestSettings/SimSpecific/$(__IMSI)">
  <Setting Name="SimSpecificValue">
    <!-- There should be one setting like this for every SIM card. -->
    <RegistrySource Type="REG_DWORD" Path="HKEY_LOCAL_MACHINE\Software\Sample\$(__IMSI)\SimSpecificValue" />
  </Setting>
</SettingsGroup>
```

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Customization answer file

10/2/2018 • 13 minutes to read • [Edit Online](#)

A **customization answer file** is an XML file that you write based on the MCSF schema. OEMs can use the MCSF customization answer file to specify the settings and variants for a custom mobile OS image. The customization answer file allows for a broader integration across the OS by providing OEMs with a single place to define nearly all mobile OS settings.

When creating or working with customization answer files, keep the following design requirements and considerations in mind:

- If you are using ImgGen.cmd to generate a mobile OS image, you can only specify one customization answer file. For more information, see [Building a mobile image using ImgGen.cmd](#).
- Depending on what you want to do, you can use the customization answer file to create a package containing your customization(s) or use the answer file as one of the inputs to create an OS image.
  - To use the customization answer file to generate a package without building an OS image, see [Generating customization packages without creating an image](#).
  - To build an image as an .ffu file using the customization answer file as one of the inputs, see [Using ImgGen.cmd to generate the image](#).
- The values used in the root customization answer file (or the one you specify as the input customization answer file during ImgGen.cmd or CustomizationGen.cmd in Step 6) is used to determine the package owner so it is important to provide values for the following attributes:
  - **Name**
  - **Description**
  - **Owner**
  - **OwnerType** (in the customization samples this is typically already set to 'OEM').
  - Any files, assets, or other settings specific to the customization
- Absolute or full paths to file sources, such as imports, assets and data files, are required. This means that whenever you specify the **Source** attribute value for a file **Import**, **Asset**, or **DataAsset**, the path must be absolute.
- File source paths may contain **\$environment\_variables**. Note that you can only use environment variables when referencing files. This will not work when configuring **Setting** values.
- Using %this\_format% in customization answer files is not allowed and results in an error.
- The source path for **Import** files may contain the macro, **\$(CurrentFileDir)**.
- **Import** source files can use environment variables in the file name. However, other source file names cannot use environment variables.

## Sample customization answer files

You can use as reference the sample OEM customization answer file located in the %WPDKCONTENTROOT%\Samples\Customization directory. This answer file shows several configured customizations and it imports two answer files that contain real connection settings data for AT&T and T-Mobile.

## Sample 1

The following sample customization answer file shows how to:

- Specify other customization answer files to import from the root answer file.
- Define **Targets** or conditions for when a variant can be applied.
- Define **Static** settings and data, which are installed for all images, regardless of the variant.
- Define settings for a **Variant**, which are applied if the associated target's conditions are met.

```
<?xml version="1.0" encoding="utf-8" ?>
<!-- Copyright (c) Microsoft Corporation. All rights reserved. -->

<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="Sample Root Customization File"
    Description="Sample customization XML. This XML contains all the valid settings that
should be correctly parsed and applied to image."
    Owner="ContosoOEM"
    OwnerType="OEM">

    <!-- This root customization answer file imports two other customization answer files. -->
    <Imports>
        <Import Source="C:\Customization\AnswerFiles\SampleCustomizationImport1.xml" />
        <Import Source="C:\Customization\AnswerFiles\SampleCustomizationImport2.xml" />
    </Imports>

    <!-- Targets define the conditions for when a variant can be applied. One target ID must be
referenced for each variant. -->

    <!-- These examples show the definition for three sample operators. -->
    <!-- In the example below, targets are set up for the variant to use in the root file. -->
    <Targets>
        <Target Id="SIM_TinyMO">
            <TargetState>
                <Condition Name="MNC" Value="26" />
                <Condition Name="MCC" Value="310" />
            </TargetState>
        </Target>
        <Target Id="SIM_BigMO">
            <TargetState>
                <Condition Name="MNC" Value="15" />
                <Condition Name="MCC" Value="310" />
            </TargetState>
        </Target>
        <Target Id="Known_BigMO">
            <TargetState>
                <Condition Name="MNC" Value="55" />
                <Condition Name="MCC" Value="310" />
            </TargetState>
        </Target>
    </Targets>

    <!-- These static settings and data will be installed for all images, regardless of the variant. -->
    <Static>
        <DataAssets Type="MapData">
            <DataAsset Source="C:\Customization\TestData\maps\OEMMap_USA.map" />
            <DataAsset Source="C:\Customization\TestData\maps\OEMMap_Canada.map" />
        </DataAssets>

        <Applications>
            <Application Source="C:\Customization\TestData\apps\OEMMOApp.xap"
                License="C:\Customization\TestData\apps\OEMMOApp_License.xml"
                ProvXML="C:\Customization\TestData\apps\MPAP_OEMMOApp_01.provxml" />
        </Applications>
    </Static>
</ImageCustomizations>
```

```

<Settings Path="TestSettingGroup">
    <Setting Name="Level1/MySetting" Value="Blue" />
    <Setting Name="MySettingAsset" Value="Alpha.jpg" />
    <Asset Name="Asset" Source="C:\Customization\MySettingAssets\Alpha.jpg" />
    <Asset Name="Asset" Source="C:\Customization\MySettingAssets\Beta.jpg" />
    <Asset Name="Asset" Source="C:\Customization\MySettingAssets\Delta.jpg" />
</Settings>

<Settings Path="TestSettingsGroup2">
    <Setting Name="OEMStaticSetting" Value="OEM Static Setting" />
</Settings>
</Static>

<!-- These settings in the Variant groups will only be applied if the associated target's conditions are met. --&gt;

<!-- The settings shown here will only be applied for the Known Big MO Variant. --&gt;
&lt;Variant Name="Known Big MO Variant"&gt;
    &lt;!-- Only one TargetRef can be used for each variant --&gt;
    &lt;TargetRefs&gt;
        &lt;TargetRef Id="Known_BigMO" /&gt;
    &lt;/TargetRefs&gt;

    &lt;Settings Path="EventSounds"&gt;
        &lt;Asset Name="Ringtones" Source="C:\Resources\Ringtones\KnownBigMO.wma" TargetFileName="BigMO.wma"
DisplayName="BigSound" Type="MobileOperator" /&gt;
        &lt;Setting Name="DefaultRingtone" Value="BigMO.wma" /&gt;
    &lt;/Settings&gt;

    &lt;Settings Path="LockScreen"&gt;
        &lt;Asset Name="Wallpapers" Source="C:\Resources\Wallpapers\KnownBigMO.jpg" TargetFileName="BigMO.jpg"
DisplayName="BigMO" Type="MobileOperator" /&gt;
        &lt;Setting Name="DefaultWallpaper" Value="BigMO.jpg" /&gt;
    &lt;/Settings&gt;
&lt;/Variant&gt;

<!-- The settings shown here will only be applied for the Tiny MO Variant. --&gt;
&lt;Variant Name="Tiny MO Variant"&gt;
    &lt;TargetRefs&gt;
        &lt;TargetRef Id="SIM_TinyMO" /&gt;
    &lt;/TargetRefs&gt;

    &lt;Settings Path="TestSettingsGroup1"&gt;
        &lt;Setting Name="Setting1" Value="Tiny MO Setting 1" /&gt;
        &lt;Setting Name="Setting3" Value="Tiny MO Setting 3" /&gt;
    &lt;/Settings&gt;

    &lt;Settings Path="EventSounds"&gt;
        &lt;Asset Name="Ringtones" Source="C:\Resources\Ringtones\TinyMO.wma" TargetFileName="TinyMO.wma"
DisplayName="TinySound" Type="MobileOperator" /&gt;
        &lt;Setting Name="DefaultRingtone" Value="TinyMO.wma" /&gt;
    &lt;/Settings&gt;
&lt;/Variant&gt;

<!-- The settings shown here will only be applied for the Big MO Variant. --&gt;
&lt;Variant Name="Big MO Variant"&gt;
    &lt;TargetRefs&gt;
        &lt;TargetRef Id="SIM_BigMO" /&gt;
    &lt;/TargetRefs&gt;

    &lt;Settings Path="TestSettingsGroup1"&gt;
        &lt;Setting Name="Setting1" Value="Big MO Setting 1" /&gt;
    &lt;/Settings&gt;
&lt;/Variant&gt;

&lt;/ImageCustomizations&gt;
</pre>

```

## Sample 2

The following sample customization answer file snippet shows how to:

- Define a **Static** variant
- Specify the customizations for a specific mobile operator in a **Variant** section.

```
<!-- Variant Section -->

<!-- Example of a static variant. There are no TargetRefs. -->

<Static>
    <!-- Asset data to be copied to the device. A small set of approved destinations are allowed.
        Only allowed in the static settings -->
    <DataAssets>
        <DataAsset Source="C:\Customization\Assets\OEMMap_USA.map" Type="MapData" />
    </DataAssets>
</Static>

<!-- Simple example of some things to do on a hypothetical TinyMO. Setting the Boot Screen
    on FirstSIM, and the MMS Gateway whenever the SIM is detected as "Newly Inserted". Ringtones and
    wallpapers are also added. -->

<Variant Name="TinyMO Settings">
    <TargetRefs>
        <TargetRef Id="SIM_TinyMO" />
    </TargetRefs>

    <Settings Path="Connectivity">
        <Setting Name="MMSGateway" Value="123.tinymo.com" />
    </Settings>

    <Settings Path="EventSounds">
        <Asset Name="Ringtones" Source="C:\Assets\Ringtones\Ringtone1.wma" />
        <Asset Name="Ringtones" Source="C:\Assets\Ringtones\Ringtone2.wma" />
        <Asset Name="Ringtones" Source="C:\Assets\Ringtones\Ringtone3.wma" />
        <Setting Name="DefaultRingtone" Value="Ringtone1.wma" />
    </Settings>

    <Settings Path="LockScreen">
        <Asset Name="Wallpapers" Source="C:\Assets\Wallpapers\Lockscreen1.jpg" />
        <Asset Name="Wallpapers" Source="C:\Assets\Wallpapers\Lockscreen2.png" />
        <Asset Name="Wallpapers" Source="C:\Assets\Wallpapers\Lockscreen3.jpeg" />
        <Setting Name="DefaultWallpaper" Value="Lockscreen2.png" />
    </Settings>

</Variant>
```

## Customization XML elements

The following table defines all the elements in a customization answer file.

ELEMENT	PARENT	DESCRIPTION
<b>Imports</b>	-	Contains the description for other customization answer files to import.

ELEMENT	PARENT	DESCRIPTION
<b>Import</b>	<b>Imports</b>	<p>Specifies the file to import and merge with the customizations in the root file.</p> <p>If the same setting is set in the imported file, that setting will be overridden by the value specified by the same in the root file.</p>
<b>Targets</b>	-	Defines the conditions for when a variant can be applied.
<b>Target</b>	<b>Targets</b>	<p>One <b>Target Id</b> must be specified for each variant. This can contain multiple sets of <b>TargetState</b> that can cause the target to fire.</p> <p>A single <b>Condition</b> is defined by a single name/value pair. The name is a runtime configuration condition name such as MNC or an OEM condition that can be defined in <b>CustomTargetState</b>. Multiple <b>Condition</b> can be contained within the <b>TargetState</b>.</p>
<b>Static</b>	-	<p>Contains the description for the static variation.</p> <p>The settings that are registry based will override values that are set in packages.</p>

ELEMENT	PARENT	DESCRIPTION
<b>DataAssets</b>	<b>Static</b>	<p>Enables the ability to put data files directly onto the data partition without a package. OEMs specify the type of data from an approved list and the image customization process determines the destination for the data based on the type.</p> <p>In Windows Phone 8.1, the following types of data are supported:</p> <ul style="list-style-type: none"> <li>• <b>MapData</b> – Use to specify map data</li> <li>• <b>RetailDemo_Microsoft</b> – Use to add custom demo content to the Microsoft retail mode provisioning app. This data asset is reserved for Microsoft use and the data assets are provided by Microsoft Marketing. Partners should contact Microsoft to reach your Microsoft marketing contact.</li> <li>• <b>RetailDemo_OEM</b> – Use to add custom demo content to the OEM retail mode provisioning app.</li> <li>• <b>RetailDemo_MO</b> – Use to add custom demo content to the mobile operator retail mode provisioning app.</li> <li>• <b>RetailDemo_Apps</b> – Use to add free or trial apps and games to be included in demo phones.</li> </ul> <p>Note that when specifying any retail demo data asset in the customization answer file, the retail mode data asset must point to a directory structure where nested files are placed in a very particular format. For more information, see the Retail Demo Mode Programmers Guide, which will be available in a future documentation release.</p>
<b>Variant</b>	-	<p>These settings in the Variant groups will only be applied if the associated target's conditions are met.</p>

ELEMENT	PARENT	DESCRIPTION
<b>TargetRefs</b>	<b>Variant</b>	<p>A collection of targets that causes the variant to be provisioned.</p> <p>Currently, only one <b>TargetRef</b> is supported.</p>
<b>Applications</b>	<b>Static, Variant</b>	<p>A collection of applications to be applied statically to the phone, or when a variant is provisioned.</p> <p>For a static variant, the app is installed into the system partition. For non-static variants, the .xap file of the app is always included in the data partition.</p> <p>For more information, see <a href="#">Apps: Preloading and storage location</a>.</p>
<b>Settings</b>	<b>Static, Variant</b>	<p>A settings group that is determined by a provided path.</p> <p>For more information about the settings group, setting elements, and associated attributes, see <a href="#">Managed Centralized Settings Framework (MCSF)</a>.</p>
<b>Setting</b>	<b>Settings</b>	<p>A single setting and value that is determined by a name. A <b>Setting</b> can contain assets or data files associated with the setting.</p> <div style="border: 1px solid black; padding: 5px;"> <p><b>Note</b></p> <p>If the policy for a specified setting is not included in a package in the image, the setting will not be included in the image and a warning will be displayed.</p> </div>

ELEMENT	PARENT	DESCRIPTION
<b>Asset</b>	<b>Settings</b>	<p>Indicates a file that needs to be included in the phone image and associated with the <b>Settings</b> to which it belongs.</p> <p>When specifying an asset, OEMs must provide the <b>Name</b> and <b>Source</b> attributes. The <b>Name</b> determines where the asset is stored on the phones. OEMs can specify multiple files using the same asset name, such as for multiple lock screen backgrounds, ringtones, and so on. The <b>Source</b> must be set to the location of the asset on your machine.</p> <p>OEMs can use the following optional attributes:</p> <ul style="list-style-type: none"> <li><b>TargetFileName</b> can be used to set the name of the file on the device. If there is no <b>TargetFileName</b> specified, the source name is used.</li> <li><b>DisplayName</b> can be used by some settings to display a name such as the ringtone's display name. The setting determines exactly how this property is used and is a passthrough to a registry value.</li> <li><b>Type</b> is a flag that indicates the type of asset, as required by some settings. In Windows Phone 8.1, <b>MO</b> and <b>OEM</b> are supported. If no value is specified, the default value is OEM.</li> </ul>

## Specifying data values in customization answer files

When creating your customization answer file, certain values must be specified in a specific format. These are as follows:

TYPE VALUE	VALUE ATTRIBUTE	DESCRIPTION
------------	-----------------	-------------

TYPE VALUE	VALUE ATTRIBUTE	DESCRIPTION
REG_DWORD	Decimal: 1 - or - Hexadecimal: 0x1	32-bit number presented in either decimal or hexadecimal format.  All hexadecimal values in the customization answer file must be prefixed with <b>0x</b> . For example, if the hexadecimal value that you need to specify is 1, you must set the <b>Value</b> in the customization answer file to <b>0x1</b> . These values are in little-endian format where the multibyte value is stored in memory from lowest byte (the "little end") to the highest byte. For example, 0x12345678 is stored as (0x78 0x56 0x34 0x12) in little-endian format.
Options defined in the policy settings files. For example:  <Validate>  <Option Value="0" FriendlyName="Red" />  <Option Value="1" FriendlyName="Green" />  <Option Value="2" FriendlyName="Blue" />  </Validate>	0 or Red 1 or Green 2 or Blue	Use the enumerated value. For example, either the enumerated value 0 or its equivalent friendly name "Red" can be specified.
REG_MULTI_SZ	Red.png;Green.png;Blue.png	Multiple text strings separated by a semicolon ';'.  A ';' is used as the delimiter for REG_MULTI_SZ so this character cannot be used inside the string because it will be parsed as the separator of two strings.
REG_BINARY	E8,03  (which corresponds to 0x03E8 hexadecimal)	Byte array in hexadecimal format, separated by a comma ','.  Do not add the 0x prefix when specifying this type of registry value in the customization answer file.

## Targets

When creating a customization answer file, OEMs can define **Targets** to describe keying for a variant. Targets for variants must be described or pre-declared before being referenced by the variant. OEMs can use the same target in multiple variants to enable reuse. Within a target, if the device meets any of the target states, the settings will be applied (the states are **OR**'ed together). The conditions within the states are **AND**'ed together and all of the conditions must be met in order for a state to be true.

In the previous XML sample, a SIM target is defined through MNC/MCC pairs. The following example shows how the SIM targets may be defined:

```
<Targets>
  <Target Id="SIM_TinyMO">
    <TargetState>
      <Condition Name="MNC" Value="123" />
      <Condition Name="MCC" Value="456" />
    </TargetState>
    <TargetState>
      <Condition Name="MNC" Value="456" />
      <Condition Name="MCC" Value="123" />
    </TargetState>
  </Target>
</Targets>
```

### Extension provisioning keys

OEMs can use custom CSPs as conditions by listing the name/path in the **Name** attribute and the desired value to pass to the CSP to check against.

## Importing other customization answer files

A customization answer file can be used to import other customization answer files that, when merged, form a single set of customizations and variants to be applied to the OS image.

### Priority order for imported customization answer files

When importing customization answer files, if there are any settings that are defined twice, the custom image will not build unless a priority within the imported answer files are specified to determine the overwrite order. To specify the overwrite order, the header for customization answer files contain an optional **Priority** attribute, which determines the order that settings will be overwritten if imported files contain a setting defined more than once. Generally, the highest priority will have its value used for customization package creation. If files have the same priority and define the same setting, the image fails to build.

To add the priority order for the imported answer file, set the **Priority** attribute within the **ImageCustomizations** block of the customization answer file to be imported. The following example shows how to do this:

```
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
  Name="Settings Input"
  Description="Settings for image build"
  Owner="Contoso"
  OwnerType="OEM"
  Priority="2">
```

### Note

When setting **Priority**, 1 is the highest, and zero (0) and negative numbers are not allowed.

### Specifying files to be imported

The following code example shows how you can specify other customization answer files to be imported. The **Imports** element must be specified in the root customization answer file.

```
<Imports>
  <Import Source="C:\Customization\SampleOperators.xml" />
  <Import Source="C:\Customization\SampleBrandCommon.xml" />
</Imports>
```

# Apps: Preloading and storage location

OEMs can preload apps using the following customization answer file code snippet:

```
<Applications>
    <Application Source="C:\Customization\ TestData\apps\OEMMOApp.xap"
        License="C:\Customization\ TestData\apps\OEMMOApp_License.xml"
        ProvXML="C:\Customization\ TestData\apps\MPAP_OEMMOApp_01.provxml" />
</Applications>
```

## Note

The AppPreInstaller is specifically looking for provXML files with the filename pattern **MPAP\_\*.\*.provxml** so make sure your file names are correctly formatted.

The following table describes where preloaded apps are stored on the device:

	FIRST OPTION	ALTERNATIVE OPTION
Default	Main OS	Data
Variant	Data	Main OS

All applications can be uninstalled by the user. When applications are uninstalled, the application files remain on the device, but these are not shown in the application list. During a cold boot, or when the user selects **Reset my phone**, apps in the Data partition will be removed while apps in the MainOS partition will be reinstalled. Users can also install apps to the SD card. When the phone boots with an SD card, the user is given an option to select the install location.

## Collisions and overrides

If there are collisions or overrides, **Settings** groups are treated as a Condition-Setting Path pair. This means that two settings using the same path but different conditions are considered unique, as are two settings with the same condition but different paths.

When two different files set the same unique **Settings** group, the following rules are used by the image customization process to resolve collisions or overrides:

- A file may overwrite a unique item that is defined by any file it imports. For example, the imports sample in the previous section, SampleDevice.customizations.xml, can be used to override the theme color set in SampleBrandCommon.xml.
- Two files imported by the same file cannot set the same unique item. For example, if SampleOperators.xml and SampleBrandCommon.xml both set the same theme color using the same condition, an error will occur and a message will be displayed to indicate that the value cannot be imported because both files define the same value.

## Related topics

[Prepare for Windows mobile development](#)

# Set phone metadata in DeviceTargetingInfo

10/2/2018 • 9 minutes to read • [Edit Online](#)

Partners are required to set certain device metadata, including hardware, support, and OEM and MO information.

Partners are required to set the following information:

- OEM and mobile operator information, used for display strings in the UI, device update, and connecting to the Microsoft Store.
- Hardware component versions and software versions, used for targeting updates to devices and for user support.
- A required phone number and optional website for user support, which appears in the **About** screen in **Settings**.

## Constraints:

- **ImageTimeOnly** – For those settings to put directly into the registry hive.
- **FirstVariationOnly** – For those settings that can be configured at runtime and potentially based on SIM value.

## Instructions

1. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="PhoneMetadataDeviceTargetingInfo"
    Description="Use to set phone metadata including the phone model name, OEM and
mobile operator name, hardware and software versions, and so on."
    Owner=""
    OwnerType="OEM">
    <!-- Define the Targets for the Variant -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>
    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>
    <Static>
        <!-- These settings are ImageTimeOnly and will be put directly into the registry hive -->
        <Settings Path="DeviceInfo/Static">
            <Setting Name="PhoneManufacturer" Value="" />
            <Setting Name="PhoneManufacturerDisplayName" Value="" />
            <Setting Name="PhoneROMVersion" Value="" />
            <Setting Name="PhoneHardwareRevision" Value="" />
            <Setting Name="PhoneSOCVersion" Value="" />
            <Setting Name="PhoneFirmwareRevision" Value="" />
            <Setting Name="PhoneRadioHardwareRevision" Value="" />
            <Setting Name="PhoneRadioSoftwareRevision" Value="" />
            <Setting Name="PhoneBootLoaderVersion" Value="" />
            <Setting Name="PhoneROMLanguage" Value="" />
            <Setting Name="PhoneHardwareVariant" Value="" />
        </Settings>
    </Static>
    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <!-- These settings are FirstVariationOnly and can be configured at runtime potentially based on
SIM value -->
        <Settings Path="DeviceInfo/Variant">
            <Setting Name="PhoneMobileOperatorName" Value="" />
            <Setting Name="PhoneManufacturerModelName" Value="" />
            <Setting Name="PhoneMobileOperatorDisplayName" Value="" />
            <Setting Name="PhoneSupportPhoneNumber" Value="" />
            <Setting Name="PhoneSupportLink" Value="" />
            <Setting Name="PhoneOEMSupportLink" Value="" />
            <Setting Name="PhoneModelName" Value="" />
            <Setting Name="RoamingSupportPhoneNumber" Value="" />
        </Settings>
    </Variant>
</ImageCustomizations>

```

2. Specify an **Owner** and configure the targets and conditions for the variant.

3. Specify a value for each of the following settings.

Settings that are **ImageTimeOnly**:

IMAGETIMEONLY SETTING NAME	DESCRIPTION
<b>PhoneManufacturer</b>	<p><b>Required.</b> This setting must contain a code specified by Microsoft that corresponds to the OEM. This setting must not be changed over time, even if an OEM name changes (such as in a merger); the original OEM name must continue to be specified.</p> <p>This setting is used for targeting device updates, for connecting to the store-within-a-store in the Microsoft Store, and for Watson reports. It also appears as part of the device friendly name on the Welcome screen, the <b>About</b> screen in <b>Settings</b>, the Ringtone list, and on the computer.</p> <p>The value must be a valid OEM ID. To get the valid OEM ID that applies to you, contact your Microsoft representative.</p> <p>The OEM ID value is in all capital letters, for example, FABRIKAM.</p> <p>To use an OEM-provided value to display in the <b>Settings &gt; About</b> screen, OEMs can use the optional <b>PhoneManufacturerDisplayName</b> setting. For more information, see the next entry in this table.</p> <p>The <b>PhoneManufacturer</b>, <b>PhoneManufacturerModelName</b>, and <b>PhoneMobileOperatorName</b> should create a unique Phone-Operator-Pairing (POP).</p>
<b>PhoneManufacturerDisplayName</b>	<p><b>Optional.</b> Use this setting to create an OEM-provided value to display in the <b>Settings &gt; About</b> screen, on apps running on the PC connected to the device through MTP, in the <b>Manufacturer</b> field in the device properties window when the device is connected to the PC through MTP, and in the list of backups on OneDrive and during initial device setup on the <b>Restore backups</b> page.</p> <p>If <b>PhoneManufacturerDisplayName</b> is set, the OS does not use the value in the <b>PhoneManufacturer</b> setting and uses the OEM-provided value to display in the <b>About</b> screen instead. If OEMs do not set a value for <b>PhoneManufacturerDisplayName</b>, the OS uses the value in the required <b>PhoneManufacturer</b> setting instead.</p> <p>When setting the value, OEMs must only use these characters: alphanumeric (A-Z, a-z, 0-9), space, period (.), comma (,)</p>
<b>PhoneROMVersion</b>	<p><b>Optional.</b> This value is specified by the silicon vendor and should not be modified by the OEM. It is used for targeting phone updates.</p> <p>This value has the format uint16.uint16.uint16.uint16.</p>

IMAGETIMEONLY SETTING NAME	DESCRIPTION
<b>PhoneHardwareRevision</b>	<p>This value is specified by the silicon vendor and should not be modified by the OEM. It is used for targeting phone updates and for Watson reports.</p> <p>This value has the format uint16.uint16.uint16.uint16.</p>
<b>PhoneSOCVersion</b>	<p>This value is specified by the silicon vendor and should not be modified by the OEM. It is used for targeting phone updates.</p> <p>This string must be less than 256 Unicode characters in length, and be alphanumeric (A-Z, a-z, 1-9). Leading and trailing spaces and other white space characters such as tabs are not permitted. The underscore character may be used to separate elements in the string, for example "Contoso_4000".</p>
<b>PhoneFirmwareRevision</b>	<p><b>Required.</b> This setting represents the complete version of the OEM software on the phone. It is used for targeting phone updates.</p> <p>This value has the format uint16.uint16.uint16.uint16. The recommended value is based on the Silicon Vendor (SV) BSP version (major.minor) and the OEM software version (major.minor), with the format "majorSV.minorSV.majorOEM.minorOEM". The value cannot be "0.0.0.0".</p>
<b>PhoneRadioHardwareRevision</b>	<p><b>Optional.</b> This value should reflect the current version of the OEM's modem hardware. It should be incremented when the modem hardware is modified. It is used for targeting phone updates.</p> <p>This string must be less than 256 Unicode characters in length, and be alphanumeric (A-Z, a-z, 1-9). Leading and trailing spaces and other white space characters such as tabs are not permitted. The underscore character may be used to separate elements in the string, for example "Contoso_4000".</p>
<b>PhoneRadioSoftwareRevision</b>	<p>This value is specified by the silicon vendor and should not be modified by the OEM. It is used for targeting phone updates.</p> <p>This value has the format uint16/string, and can contain a maximum of 15 characters.</p>
<b>PhoneBootLoaderVersion</b>	<p><b>Optional.</b> This value is specified by the silicon vendor and should not be modified by the OEM. It is used for targeting phone updates and for Watson reports.</p>
<b>PhoneROMLanguage</b>	<p><b>Required.</b> Set the value to a four character Language Code Identifier (LCID), such as "0409" for English (US).</p>

IMAGETIMEONLY SETTING NAME	DESCRIPTION
<b>PhoneHardwareVariant</b>	<p><b>Required.</b> Use to describe the specific hardware configuration used for a particular phone model. The hardware configuration that makes up the <b>PhoneHardwareVariant</b> includes specific hardware parts such as the applications processor, radio (network bands), sensors, memory configuration, and so on. For example, if Contoso (a fictional OEM) produces a popular phone model called Fabrikam 2000 that ships on multiple mobile networks, the phone may come in two hardware variants: "VAR-CDMA", for use on CDMA networks, and "VAR-GSM" for use on GSM networks.</p> <p>Use a string value to specify the variant of the OEM's hardware. This setting is mandatory for code signing and registration.</p> <div data-bbox="815 698 1390 893" style="border: 1px solid black; padding: 10px;"> <p><b>Note</b></p> <p>OEMs creating a runtime configuration image cannot span across multiple <b>PhoneHardwareVariants</b>.</p> </div>

Settings that are **FirstVariationOnly**:

FIRSTVARIATIONONLY SETTING NAME	DESCRIPTION
<b>PhoneMobileOperatorName</b>	<p><b>Required.</b> This setting is used for targeting phone updates. It must contain a code specified by Microsoft that corresponds to the mobile operator. These codes are provided in <a href="#">Registry values for mobile operator IDs</a>. For open market phones, in which the mobile operator is not known, use the codes in <a href="#">Registry values for carrier-unlocked phones</a> instead.</p> <p>This string is not visible to the user.</p> <p>This setting must not be changed over time even if the user switches SIMs or mobile operators, as updates are always targeted based on the first mobile operator associated with the phone.</p> <p>The <b>PhoneManufacturer</b>, <b>PhoneManufacturerModelName</b>, and <b>PhoneMobileOperatorName</b> should create a unique Phone-Operator-Pairing (POP).</p>

FIRST VARIATION ONLY SETTING NAME	DESCRIPTION
<b>PhoneManufacturerModelName</b>	<p><b>Required.</b> This setting is used for targeting phone updates. It must contain a code that is registered with Microsoft to correspond to the phone model. This string must be unique – if there are any hardware differences between phones that require changes to the BSP, the phones must have different <b>PhoneManufacturerModelName</b> values. This string must not be changed after the phone is sold.</p> <p>This string is for OEM reference and can be set to any value that meets the following requirements:</p> <ul style="list-style-type: none"> <li>• The string length must be less than 256 characters</li> <li>• The string must be alphanumeric (A-Z, a-z, 1-9)</li> <li>• Leading and trailing spaces are not permitted and will cause update failures</li> </ul> <div style="border: 1px solid black; padding: 10px;"> <p><b>Note</b></p> <p>This string is returned by the <b>DeviceStatus.DeviceName</b> property. For more information, see the SDK Documentation.</p> </div> <p>Microsoft recommends using a different value for <b>PhoneManufacturerModelName</b> and another value for <b>PhoneModelName</b>. <b>PhoneManufacturerModelName</b> should be as unique as possible for the particular device revision or variant.</p> <p>The <b>PhoneManufacturer</b>, <b>PhoneManufacturerModelName</b>, and <b>PhoneMobileOperatorName</b> should create a unique Phone-Operator-Pairing (POP).</p>
<b>PhoneMobileOperatorDisplayName</b>	<p><b>Optional.</b> Defines the friendly name of the Mobile Operator. This string is displayed in the support section of the <b>About</b> screen in <b>Settings</b> and in the ringtone list.</p>
<b>PhoneSupportPhoneNumber</b>	<p><b>Optional.</b> Specifies the OEM or mobile operator's support contact phone number. This string is displayed in the <b>About</b> screen in <b>Settings</b>. This setting also corresponds to the Genuine Windows Phone Certificates (GWPC) support number. This should be a string of numbers. The country code is not required. This setting varies by partner.</p>

FIRST VARIATION ONLY SETTING NAME	DESCRIPTION
<b>PhoneSupportLink</b>	<p><b>Optional.</b> Specifies the mobile operator's support website. The default is an empty string (""), which means that a support link will not be displayed to the user.</p> <p>This should be a functional link that starts with http://. The link should be a URL that redirects to the mobile version of the web page. The content in the webpage should reflow to the screen width. This can be achieved by adding the CSS Tag &amp;quot;@-ms-viewport { width: device-width; }"</p> <p>This setting varies by mobile operator.</p>
<b>PhoneOEMSupportLink</b>	<p><b>Optional.</b> Specifies the OEM's support website. The default is an empty string (""), which means that a support link will not be displayed to the user.</p> <p>This should be a functional link that starts with http://. The link should be a URL that redirects to the mobile version of the web page. The content in the webpage should reflow to the screen width. This can be achieved by adding the CSS Tag &amp;quot;@-ms-viewport { width: device-width; }"</p> <p>This setting varies by OEM.</p>

FIRST VARIATION ONLY SETTING NAME	DESCRIPTION
<b>PhoneModelName</b>	<p><b>Required.</b> This string is the brand name of the phone and is used for Customer Support and Watson reports. This string is the name marketed to consumers or end users and appears as part of the phone friendly name in multiple places in the phone's UI including: on the Welcome screen, the <b>About</b> screen in <b>Settings</b>, and on the desktop computer.</p> <p>OEMs should ensure that this value matches the value of the <b>ProductString</b> value under the <b>HKEY_LOCAL_MACHINE</b> registry subtree's <b>CurrentControlSet\Control\USBFN\Default</b> subkey.</p> <p>Microsoft recommends that partners consider the following when specifying the value for the <b>PhoneModelName</b> setting:</p> <ul style="list-style-type: none"> <li>• Leave this name blank during development until the phone is ready to enter trials, as applications can collect and read this value.</li> <li>• Do not include the manufacturer or OEM name when setting the value for this setting. There are dialogs on the phone that display <b>PhoneManufacturer</b> appended to <b>PhoneModelName</b> so including the OEM or manufacturer's name when setting <b>PhoneModelName</b> will result in repetitions. For example, if <b>PhoneManufacturer</b> is set to <i>Contoso</i> and <b>PhoneModelName</b> is set to <i>Contoso Phone z102</i>, the result will show up as <i>Contoso Contoso Phone z102</i>.</li> <li>• Use a different value for <b>PhoneModelName</b> and another value for <b>PhoneManufacturerModelName</b>. The latter should be as unique as possible for the particular device revision or variant.</li> </ul>
<b>RoamingSupportPhoneNumber</b>	<p><b>Optional.</b> Specifies the OEM or mobile operator's roaming support contact phone number. This string is displayed in the <b>About</b> screen in <b>Settings</b>.</p> <p>For C+G dual SIM phones, OEMs may need to configure this setting. For more information, see <a href="#">Configure C+G dual SIM settings</a>.</p>

## Testing steps

1. Flash the build containing this customization to a phone.
2. Go to the **About** screen in **Settings**. Tap on the **More Info** button.
3. Verify that the information on this screen matches the values you specified.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Set languages and locales

10/2/2018 • 8 minutes to read • [Edit Online](#)

There are several types of language settings that partners can control on the mobile device. The languages that partners choose to include should be based on the market in which the device will ship, and the amount of space available for language data. The following table shows the different kind of language and locale settings. This is just an overview. For more detailed information, including implementation instructions and limitations and restrictions, click through to the following topics.

	<b>PARTNERS MUST SPECIFY</b>	<b>HOW MANY</b>	<b>USER IMPACT</b>
<a href="#">Mobile device languages</a>	<ul style="list-style-type: none"><li>The set of available mobile device languages that need to be installed. OEMs must specify the value they want to use using the <b>UserInterface</b> element in the OEMInput.xml file.</li><li>The default mobile device language. OEMs can specify this value using the <b>BootUILanguage</b> element in the OEMInput.xml file.</li></ul>	<ul style="list-style-type: none"><li>At least one. The maximum number is determined by space constraints.</li><li>One default device language is required.</li></ul>	There is no way to add additional languages to the device later without re-flashing a new image.
<a href="#">Regional format</a>	The OEM must specify a default locale, which determines the country or region, regional format, pre-enabled keyboard, and speech languages. OEMs can specify the value this value using the <b>BootLocale</b> element in the OEMInput.xml file.	One locale is required.	The user can change the country/region.
<a href="#">Keyboard layout</a>	Nothing. 160 keyboard layouts are included on the device by default, and OEMs cannot modify, add, or delete keyboard layouts from the OS image.		

	PARTNERS MUST SPECIFY	HOW MANY	USER IMPACT
Text correction and suggestions	The set of keyboard language files used for text correction and suggestions while typing. OEMs must specify the value they want to use using the <b>Keyboard</b> element in the OEMInput.xml file.	At least one. The maximum number is determined by space constraints.	Users can download additional keyboard language files as needed. If the keyboard language file for a given keyboard is not already on the device, it is downloaded automatically when the user enables the keyboard for the first time.
Pre-enabled keyboard	<p>For each mobile device language, the OS automatically determines which keyboards to show the user. If the OEM needs to enable additional keyboards by default to meet the needs of their market, they can.</p> <p>The <b>Recommended additional pre-enabled keyboard(s)</b> apply to these locales:</p> <ul style="list-style-type: none"> <li>• The primary language script is non-Latin or where the default keyboard is a Latin-based keyboard so users can type their Microsoft account email and password.</li> <li>• There is more than one official language.</li> </ul>	Additional keyboards are not necessary for most markets.	Users can select which keyboards they wish to use.
Speech languages	The set of available speech languages on the mobile device. OEMs can specify the value they want to use using the <b>Speech</b> element in the OEMInput.xml file.	OEMs are not required to include speech by default. The maximum number is determined by space constraints.	Users can download additional speech languages if they need them.

The following table shows recommended default mobile device languages and regional formats for different markets. The default keyboard language is automatically generated by the OS based on the device language. Each default keyboard language has at least one keyboard associated with it that is enabled by default. The speech language specified in the final column will only be enabled if the OEM has included it in the OS image or the user has downloaded it.

#### WARNING

If you set the **BootUILanguage** and **BootLocale** to a pair of languages that are not recommended in the following table, the OS image that will be created will lead to an inconsistent post-setup experience where the regional format doesn't follow the language selection. Microsoft strongly recommends that OEMs follow the recommended languages outlined in this table.

Market	Default Mobile Device Language	Default Regional Format	Default Keyboard Language Enabled by the Device	Keyboard Values to Specify in OEMINPUT.XML	Recommended Additional Pre-Enabled Keyboard(s)	Speech Values to Specify in OEMINPUT.XML
Albania	sq-AL	sq-AL	sq-AL	en-GB	none	none
Algeria	ar-SA	ar-DZ	en-US	ar-SA, en-US	ar-SA	none
Argentina	es-MX	es-AR	es-MX	es-MX	none	none
Australia	en-GB	en-AU	en-GB	en-GB	none	none
Austria	de-DE	de-AT	de-DE	de-DE	none	none
Azerbaijan	az-Latn-AZ	az-Latn-AZ	az-Latn-AZ	en-GB	none	none
Bahrain	ar-SA	ar-BH	en-US	ar-SA, en-US	ar-SA	none
Bangladesh	bn-BD	bn-BD	en-GB	en-GB	bn-BD	none
Belarus	be-BY	be-BY	en-US	en-US	be-BY	none
Belgium	fr-FR	fr-BE	fr-FR	fr-FR	none	none
Belgium	nl-NL	nl-BE	nl-BE	nl-BE	none	none
Belize	en-GB	en-BZ	en-GB	en-GB	none	none
Bolivarian Republic of Venezuela	es-MX	es-VE	es-MX	es-MX	none	none
Bolivia	es-MX	es-BO	es-MX	es-MX	none	none
Bosnia and Herzegovina (Croatian)	hr-HR	hr-BA	hr-HR	hr-HR	none	none
Bosnia and Herzegovina (Serbian)	sr-Latn-RS	sr-Latn-BA	sr-Latn-RS	sr-Latn-CS	none	none
Brazil	pt-BR	pt-BR	pt-BR	pt-BR	none	pt-BR
Brunei	ms-MY	ms-BN	ms-MY	ms-MY	ms-MY	none
Bulgaria	bg-BG	bg-BG	en-US	en-US	bg-BG	none
Cambodia	km-KH	km-KH	en-US	en-US	km-KH	none
Cameroon	fr-FR	fr-CM	fr-FR	fr-FR	none	none

Market	Default Mobile Device Language	Default Regional Format	Default Keyboard Language Enabled by the Device	Keyboard Values to Specify in OEMINPUT.XML	Recommended Additional Pre-Enabled Keyboard(s)	Speech Values to Specify in OEMINPUT.XML
Canada (English)	en-GB	en-CA	en-US	en-US	none	none
Canada (French)	fr-CA	fr-CA	fr-CA	fr-CA	none	none
Chile	es-MX	es-CL	es-MX	es-MX	none	none
China	zh-CN	zh-CN	zh-CN	zh-CN	none	zh-CN
Colombia	es-MX	es-CO	es-MX	es-MX	none	none
Costa Rica	es-MX	es-CR	es-MX	es-MX	none	none
Cote d'Ivoire	fr-FR	fr-Cl	fr-FR	fr-FR	none	none
Croatia	hr-HR	hr-HR	hr-HR	en-GB	none	none
Czech Republic	cs-CZ	cs-CZ	cs-CZ	cs-CZ	none	none
Denmark	da-DK	da-DK	da-DK	da-DK	none	none
Dominican Republic	es-MX	es-DO	es-MX	es-MX	none	none
Ecuador	es-MX	es-EC	es-MX	es-MX	none	none
Egypt	ar-SA	ar-EG	en-US	ar-SA, en-US	ar-SA	none
El Salvador	es-MX	es-SV	es-MX	es-MX	none	none
Estonia	et-EE	et-EE	et-EE	en-GB	none	none
Ethiopia	am-ET	am-ET	am-ET	am-ET	none	none
Finland (Finnish)	fi-FI	fi-FI	fi-FI	fi-FI	none	none
Finland (Swedish)	sv-SE	sv-FI	sv-SE	sv-SE	none	none
France	fr-FR	fr-FR	fr-FR	fr-FR	none	fr-FR
Germany	de-DE	de-DE	de-DE	de-DE	none	de-DE
Greece	el-GR	el-GR	en-US	el-GR, en-US	el-GR	none
Guatemala	es-MX	es-GT	es-MX	es-MX	none	none

Market	Default Mobile Device Language	Default Regional Format	Default Keyboard Language Enabled by the Device	Keyboard Values to Specify in OEMINPUT.XML	Recommended Additional Pre-Enabled Keyboard(s)	Speech Values to Specify in OEMINPUT.XML
Haiti	fr-FR	fr-HT	fr-FR	fr-FR	none	none
Honduras	es-MX	es-HN	es-MX	es-MX	none	none
Hong Kong S.A.R. (Chinese)	zh-TW	zh-HK	en-GB	zh-HK, en-GB	zh-HK	zh-HK
Hong Kong S.A.R. (English)	en-GB	en-HK	en-GB	en-GB	en-GB	none
Hungary	hu-HU	hu-HU	hu-HU	hu-HU	none	none
Iceland	is-IS	is-IS	is-IS	is-IS	none	none
India (English)	en-GB	en-IN	en-IN	en-IN	hi	en-IN
India (Hindi)	hi-IN	hi-IN	en-IN	hi-IN, en-IN	hi, hi-IN	none
Indonesia	id-ID	id-ID	id-ID	id-ID	none	none
Iran	fa-IR	fa-IR	en-US	fa-IR, en-US	fa-IR	none
Iraq	ar-SA	ar-IQ	en-US	ar-SA, en-US	ar-SA	none
Ireland	en-GB	en-IE	en-GB	en-GB	none	none
Israel	he-IL	he-IL	en-US	he-IL, en-US	he-IL	none
Italy	it-IT	it-IT	it-IT	it-IT	none	it-IT
Jamaica	en-GB	en-JM	en-GB	en-GB	none	none
Japan	ja-JP	ja-JP	ja-JP	ja-JP	none	ja-JP
Jordan	ar-SA	ar-JO	en-US	ar-SA, en-US	ar-SA	none
Kazakhstan	kk-KZ	kk-KZ	en-US	en-US	kk-KZ	none
Kenya (Kiswahili)	sw-KE	sw-KE	sw-KE	sw-KE	sw-KE	none
Korea	ko-KR	ko-KR	en-US	ko-KR, en-US	ko-KR	none
Kuwait	ar-SA	ar-KW	en-US	ar-SA, en-US	ar-SA	none
Laos	lo-LA	lo-LA	en-US	en-US	lo-LA	none

Market	Default Mobile Device Language	Default Regional Format	Default Keyboard Language Enabled by the Device	Keyboard Values to Specify in OEMINPUT.XML	Recommended Additional Pre-Enabled Keyboard(s)	Speech Values to Specify in OEMINPUT.XML
Latvia	lv-LV	lv-LV	lv-LV	en-GB	none	none
Lebanon	ar-SA	ar-LB	en-US	ar-SA, en-US	ar-SA	none
Liechtenstein	de-DE	de-LI	de-DE	de-DE	none	none
Lithuania	lt-LT	lt-LT	lt-LT	en-GB	none	none
Luxembourg (French)	fr-FR	fr-LU	fr-FR	fr-FR	none	none
Luxembourg (German)	de-DE	de-LU	de-DE	de-DE	none	none
Macao S.A.R.	zh-TW	zh-MO	en-GB	zh-TW, en-GB	zh-TW	none
Macedonia, FYRO	mk-MK	mk-MK	en-US	en-US	mk-MK	none
Malaysia (English)	en-GB	en-MY	en-GB	en-GB	none	none
Malaysia (Malay)	ms-MY	ms-MY	ms-MY	ms-MY	none	none
Mexico	es-MX	es-MX	es-MX	es-MX	none	es-MX
Moldova	ro-RO	ro-MD	ro-RO	ro-RO	none	none
Monaco	fr-FR	fr-MC	fr-FR	fr-FR	none	none
Montenegro	sr-Latn-RS	sr-Latn-ME	sr-Latn-CS	en-GB	none	none
Morocco (Arabic)	ar-SA	ar-MA	fr-FR	ar-SA, en-US	ar-SA	none
Morocco (French)	fr-FR	fr-MA	fr-FR	fr-FR	none	none
Netherlands	nl-NL	nl-NL	nl-NL	nl-NL	none	none
New Zealand	en-GB	en-NZ	en-GB	en-GB	none	none
Nicaragua	es-MX	es-NI	es-MX	es-MX	none	none
Nigeria	ha-Latn-NG	ha-Latn-NG	ha-Latn-NG	ha-Latn-NG	none	none
Norway	nb-NO	nb-NO	nb-NO	nb-NO	none	none

Market	Default Mobile Device Language	Default Regional Format	Default Keyboard Language Enabled by the Device	Keyboard Values to Specify in OEMINPUT.XML	Recommended Additional Pre-Enabled Keyboard(s)	Speech Values to Specify in OEMINPUT.XML
Oman	ar-SA	ar-OM	en-US	ar-SA, en-US	ar-SA	none
Panama	es-MX	es-PA	es-MX	es-MX	none	none
Paraguay	es-MX	es-PY	es-MX	es-MX	none	none
Peru	es-MX	es-PE	es-MX	es-MX	none	none
Philippines (Filipino)	fil-PH	fil-PH	en-US	en-US	none	none
Philippines (English)	en-US	en-PH	en-US	en-US	none	none
Poland	pl-PL	pl-PL	pl-PL	pl-PL	none	pl-PL
Portugal	pt-PT	pt-PT	pt-PT	pt-PT	none	none
Puerto Rico	es-MX	es-PR	es-MX	es-MX	none	none
Qatar	ar-SA	ar-QA	en-US	ar-SA, en-US	ar-SA	none
Reunion	fr-FR	fr-RE	fr-FR	fr-FR	none	none
Romania	ro-RO	ro-RO	ro-RO	ro-RO	none	none
Russia	ru-RU	ru-RU	en-US	ru-RU, en-US	ru-RU	ru-RU
Saudi Arabia	ar-SA	ar-SA	en-US	ar-SA, en-US	ar-SA	none
Senegal	fr-FR	fr-SN	fr-FR	fr-FR	none	none
Serbia	sr-Latn-RS	sr-Latn-RS	sr-Latn-CS	en-GB	none	none
Singapore (Chinese)	zh-CN	zh-SG	en-GB	zh-CN, en-GB	zh-CN	none
Singapore (English)	en-GB	en-SG	en-GB	en-GB	none	none
Slovakia	sk-SK	sk-SK	sk-SK	sk-SK	none	none
Slovenia	sl-SI	sl-SI	sl-SI	en-GB	none	none
South Africa (Afrikaans)	af-ZA	af-ZA	af-ZA	af-ZA	none	none
South Africa (English)	en-GB	en-ZA	en-GB	en-GB	none	none

Market	Default Mobile Device Language	Default Regional Format	Default Keyboard Language Enabled by the Device	Keyboard Values to Specify in OEMINPUT.XML	Recommended Additional Pre-Enabled Keyboard(s)	Speech Values to Specify in OEMINPUT.XML
Spain (Basque)	eu-ES	eu-ES	eu-ES	eu-ES	none	none
Spain (Catalan)	ca-ES	ca-ES	ca-ES	ca-ES	none	none
Spain (Galician)	gl-ES	gl-ES	gl-ES	gl-ES	none	none
Spain (Spanish)	es-ES	es-ES	es-ES	es-ES	none	es-ES
Sweden	sv-SE	sv-SE	sv-SE	sv-SE	none	none
Switzerland (French)	fr-FR	fr-CH	fr-CH	fr-CH	none	none
Switzerland (German)	de-DE	de-CH	de-DE	de-DE	none	none
Switzerland (Italian)	it-IT	it-CH	it-IT	it-IT	none	none
Syria	ar-SA	ar-SY	en-US	ar-SA, en-US	ar-SA	none
Taiwan	zh-TW	zh-TW	en-US	zh-TW, en-US	zh-TW	zh-TW
Thailand	th-TH	th-TH	en-US	en-US	th-TH	none
Trinidad and Tobago	en-GB	en-TT	en-GB	en-GB	none	none
Tunisia	ar-SA	ar-TN	en-US	ar-SA, en-US	ar-SA	none
Turkey	tr-TR	tr-TR	tr-TR	tr-TR	tr-TR	none
Ukraine	uk-UA	uk-UA	en-US	en-US	uk-UA	none
United Arab Emirates	ar-SA	ar-AE	en-US	ar-SA, en-US	ar-SA	none
United Kingdom	en-GB	en-GB	en-GB	en-GB	none	en-GB
United States (English)	en-US	en-US	en-US	en-US	none	en-US
United States (Spanish)	es-MX	es-US	es-MX	es-MX	none	none

MARKET	DEFAULT MOBILE DEVICE LANGUAGE	DEFAULT REGIONAL FORMAT	DEFAULT KEYBOARD LANGUAGE ENABLED BY THE DEVICE	KEYBOARD VALUES TO SPECIFY IN OEMINPUT.XML	RECOMMENDED ADDITIONAL PRE-ENABLED KEYBOARD(S)	SPEECH VALUES TO SPECIFY IN OEMINPUT.XML
Uruguay	es-MX	es-UY	es-MX	es-MX	none	none
Uzbekistan	uz-Latn-UZ	uz-Latn-UZ	uz-Latn-UZ	en-GB	none	none
Vietnam	vi-VN	vi-VN	vi-VN	vi-VN	none	none
Yemen	ar-SA	ar-YE	en-US	ar-SA, en-US	ar-SA	none
Zimbabwe	en-GB	en-ZW	en-GB	en-GB	none	none

# Create a resource-only .dll for localized strings

10/2/2018 • 5 minutes to read • [Edit Online](#)

When you have multiple display languages included on a phone, you must create a resource-only .dll to store all of the necessary localized display strings. From the .dll and its associated .dll.mui files, you can access strings in the current language from the registry. This technique is used in some customizations where you can use localized display strings as input.

Note that a single common resource-only .dll file and set of .dll.mui files can be used to manage the localized display strings for all of these customizations. It is not necessary to include a separate .dll file for each customization.

## Creating a resource-only .dll for localized strings

1. In Visual Studio, create a Visual C++ Win32 project:
  - a. Click **File** > **New** > **Project**.
  - b. In the left column, expand the **Visual C++** templates, then select **Win32**.
  - c. In the center column, choose **Win32 Project**. Give your project a name according to the language you are using so you can easily identify it, such as *DisplayStrings0409*. An alternative name can be *DisplayStrings\_en\_us*.
  - d. Click **OK**.
2. In the Win32 Application Wizard:
  - a. Click **Next** on the first screen.
  - b. Select **DLL** as the **Application type**.
  - c. Uncheck **Security Development Lifecycle (SDL) checks**. Leave **ATL** and **MFC** unchecked.
  - d. Check **Empty Project**.
  - e. Click **Finish**.
3. Click on the **Project** menu and select **Properties**.
4. In the new project's Property Pages window:
  - a. Navigate to **Configuration Properties** > **Linker** > **Advanced**.
  - b. In the right column, change **No Entry Point** to **Yes (/NOENTRY)**.

/NOENTRY prevents the linker from linking a reference to \_main into the .dll; this option is required to create a resource-only .dll.
5. Without closing the project's Property Pages window:
  - a. Navigate to **Configuration Properties** > **Resources** > **Command Line**.
  - b. In the right column, locate **Additional Options** and add **/n** to the edit box.

/n specifies that each entry in the string table must be null-terminated. This prevents the entire contents of the string table from being displayed on the phone's screen in the event of an error.

- c. Click **OK**.
6. Add a new string table:
- a. Click **Project > Add Resource**.
  - b. In the Add Resource dialog, select **String Table**.
  - c. Click **New**.
7. In the string table editor, add a string resource for every string that you want to display in the Windows Phone UI.
- For each localizable string that you have added, create a row in your string table. In the **Caption** field of each row, type the localizable string for the display language.
8. Build the .dll by selecting **Build Solution** from the **Build** menu.
- This step will produce a .dll called DisplayStrings0409.dll.
9. For every additional localized display language, repeat steps 1-8.

#### **Add custom build steps to each project to split the various language resource modules into separate .mui files**

A multi-part process is involved in splitting the DLLs into one executable DisplayStrings.dll, plus a DisplayStrings.dll.mui for each of the languages that you built.

To sort the localizable resources into separate .mui files, you can add the following custom build steps to each project by opening the project properties and navigating to **Configuration Properties > Build Events**. For example, for a debug build for English (US):

1. Pre-build:

```
rmdir /s /q ".\en-us"
```

2. Pre-link:

```
mkdir ".\en-us"
```

3. Post-build line 1:

```
muirct.exe -q "DoReverseMuiLoc.rcconfig" -v 2 -x 0x0409 -g 0x0409 ".\Debug\DisplayStrings0409.dll"
".\Debug\DisplayStrings.dll" ".\en-us\DisplayStrings.dll.mui"
```

DoReverseMuiLoc.rcconfig is a type of configuration file typically used by muirct.exe to split resources between the language-neutral DLL and the language-dependent .mui files. See the next section for more information about this file.

4. Post-build line 2:

```
muirct.exe -c ".\Debug\DisplayStrings.dll" -e ".\en-us\DisplayStrings.dll.mui"
```

muirct.exe embeds a MUI resource into the DisplayStrings.dll module during splitting. To properly load at run-time the appropriate resources from the language-specific DisplayStrings.dll.mui modules, each .mui file must have its checksums fixed-up to match the checksums in the baseline language-neutral module. This is done using the command specified in this step.

Note that the separate build steps shown in this section is just one of the ways you can do this. You can also put all the steps into a post-build step. The actual commands use Visual Studio macros to target the correct output directories.

5. Repeat steps 1-4 for the other languages that you are using and replace the language-specific settings with the ones that correspond to the other languages. For example, if you are processing the German settings next:

- a. Replace `-x 0x0409` with the new language ID that you are using, such as `-x 0x0407` for German.

- b. Replace `en-us` with `de-de`.

- c. Change the output file in the post-build step from `DisplayStrings.dll` to `DisplayStrings_discard.dll`.

Once all the projects have been built, deploy the projects to your image and verify the strings are properly loaded for each language.

`DoReverseMuiLoc.rcconfig` is an XML file that contains the following lines. Copy this XML file to the `ProjectRootDirectory\ProjectName` folder.

```
<?xml version="1.0" encoding="utf-8"?>
<localization>
    <resources>
        <win32Resources fileType="Application">
            <neutralResources>
            </neutralResources>
            <localizedResources>
                <resourceType typeNameId="#1"/>
                <resourceType typeNameId="#10"/>
                <resourceType typeNameId="#1024"/>
                <resourceType typeNameId="#11"/>
                <resourceType typeNameId="#12"/>
                <resourceType typeNameId="#13"/>
                <resourceType typeNameId="#14"/>
                <resourceType typeNameId="#15"/>
                <resourceType typeNameId="#16"/>
                <resourceType typeNameId="#17"/>
                <resourceType typeNameId="#18"/>
                <resourceType typeNameId="#19"/>
                <resourceType typeNameId="#2"/>
                <resourceType typeNameId="#20"/>
                <resourceType typeNameId="#2110"/>
                <resourceType typeNameId="#23"/>
                <resourceType typeNameId="#240"/>
                <resourceType typeNameId="#3"/>
                <resourceType typeNameId="#4"/>
                <resourceType typeNameId="#5"/>
                <resourceType typeNameId="#6"/>
                <resourceType typeNameId="#7"/>
                <resourceType typeNameId="#8"/>
                <resourceType typeNameId="#9"/>
                <resourceType typeNameId="HTML"/>
                <resourceType typeNameId="MOFDATA"/>
            </localizedResources>
        </win32Resources>
    </resources>
</localization>
```

## Accessing localized strings from a customization

To add the resource .dll files to a phone image and access localized strings in a customization, follow these instructions.

1. Add the DisplayStrings.dll file and the DisplayStrings.dll.LCID.mui files to your customization answer file by using the following syntax.

```
<Settings Path="Localization/MUI">
    <!-- Use to add your base MUI DLL file -->
    <Asset Name="BaseDll" Source="C:\Path\DisplayStrings.dll" />

    <!-- Use to specify the language MUI packages (*.dll.mui) for the languages you are supporting and have localized strings for -->
    <Asset Name="LanguageDll/${langid}" Source="C:\Path\DisplayStrings.dll.mui" />
    <!-- Add as many as you need -->
</Settings>
```

Make the following changes to this XML:

- In the `BaseDll` asset, replace `C:\Path` with the location of `DisplayStrings.dll` on your development machine.
  - Add additional assets for the language MUI packages (`*.dll.mui`) for all the languages you are supporting and have localized strings for. Set each asset's `Name` to `LanguageDll/ ${langid}` where `$(langid)` corresponds to the language, such as `LanguageDll/en-US`. Also set each asset's `Source` to the location of the `.dll.mui` file for that language, such as `C:\Path\en-us\DisplayStrings.dll.mui`.
2. To access strings from a resource .dll file in a customization asset, set the `DisplayName` attribute for the asset to the name of the resource-only .dll file and specify the string offset as shown in the following customization answer file excerpt.

```
<!-- Use to add one additional alarm sound -->
<Settings Path="EventSounds">
    <Asset Name="AlarmSounds" DisplayName="@DisplayStrings.dll,-101" Source="C:\Path\NewAlarmSound.wma"
    />
</Settings>
```

This particular example shows how to access string 101 (as specified in the `Value` field of the string table) for an additional alarm sound.

# Customizations for device management

10/2/2018 • 2 minutes to read • [Edit Online](#)

This section provides more information about device management settings that OEMs can change.

## In this section

TOPIC	DESCRIPTION
<a href="#">Enabling runtime configuration</a>	<p>Runtime configuration, or multivariant, provides a generic mechanism for creating a single image that can work for multiple markets and reduce the number of images that OEMs need to create, manage, and test. By default, runtime configuration is enabled and ADC is turned off. Only one or the other must be turned on. So, if an OEM wants to disable runtime configuration, they must enable ADC.</p> <p>The OS will handle different scenarios depending on whether runtime configuration has been enabled so OEMs should take into consideration the scenarios they are trying to enable.</p>
<a href="#">Managing runtime configuration data</a>	<p>OEMs can configure the following settings to manage the cleanup of runtime configuration data on the mobile device.</p>
<a href="#">Override the default CountryTable.xml</a>	<p>The mobile runtime configuration package includes a built-in XML file (CountryTable.xml) for mapping between GeolIDs, iso3166 country/region codes, and MCCs. Windows uses this table to assist in validating 3-digit MNCs for specific countries/regions, which is done by including an MNC list for that country/region. Otherwise, the OS assumes that MNCs are valid if they are 2 digits.</p> <p>OEMs can override the default country/region lookup table and instruct the runtime configuration engine to use an OEM-provided mapping table instead.</p>
<a href="#">Setting the UICC slot for branding configuration</a>	<p>OEMs can specify which UICC slot will be used for branding configuration.</p>

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Enabling runtime configuration

10/2/2018 • 2 minutes to read • [Edit Online](#)

Runtime configuration, or multivariant, provides a generic mechanism for creating a single image that can work for multiple markets and reduce the number of images that OEMs need to create, manage, and test. By default, runtime configuration is enabled and ADC is turned off. Only one or the other must be turned on. So, if an OEM wants to disable runtime configuration, they must enable ADC.

The OS will handle different scenarios depending on whether runtime configuration has been enabled so OEMs should take into consideration the scenarios they are trying to enable.

## Note

By enabling runtime configuration, SIM-based language detection will also be enabled.

## Constraints: ImageTimeOnly

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="EnableRuntime configuration"
    Description="Use to enable runtime configuration."
    Owner=""
    OwnerType="OEM">

    <Static>
        <!-- Turns on runtime configuration -->
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <!-- Turns off ADC -->
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.
3. The above code example shows the default setting. Only one can be turned on (or enabled), and the other must be turned off (or disabled). Depending on what you want to do, you can use one of the following values:

VALUE	DESCRIPTION
0	Disables the feature
1	Enables the feature

The registry key for enabling or disabling runtime configuration is below. This registry key can have a value of 1 or 0 where 1 represents enabled and 0 represents disabled.

```
HKLM\software\microsoft\multivariant\enable
```

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Managing runtime configuration data

10/2/2018 • 5 minutes to read • [Edit Online](#)

When the runtime configuration engine applies a variant to a device, a number of assets are used and these can include maps, apps, wallpapers, ringtones, and so on. The data for these features can be significantly large. To enable many variants to ship in a single device image, multiple large sets of this data are included somewhere in storage. Only Retail Mode content, map data, and app installers are stored in the user store. Other smaller variant data is automatically placed in the OS partition.

To allow users to reset their device and not wait for apps to download from the Microsoft Store if the same variant is used, the OS protects the data by copying it to the OS partition. The following table describes what happens to the device content during initial install, upon resetting the storage limit, and after the device is reset.

CONTENT	INITIAL STORAGE LOCATION	INITIAL STORAGE LOCATION	RESULT UPON RESET/STORAGE LIMIT	RESULT UPON RESET/STORAGE LIMIT	RESULT AFTER RESET	RESULT AFTER RESET
	<b>Selected subvariant</b>	<b>Other subvariants</b>	<b>Selected subvariant</b>	<b>Other subvariants</b>	<b>If the same subvariant is selected</b>	<b>Other subvariants</b>
UI languages	OS partition	OS partition	Stays in OS partition	Stays in OS partition	Used from OS partition	Used from OS partition
Retail mode	User partition	User partition	Deleted	Deleted	Downloaded from the Internet	Downloaded from the Internet
Applications	User partition	User partition	Copied to OS partition	Deleted	Used from OS partition	Downloaded from the Internet
Wallpapers	OS partition	OS partition	Stays in OS partition	Stays in OS partition	Used from OS partition	Used from OS partition
Ringtones	OS partition	OS partition	Stays in OS partition	Stays in OS partition	Used from OS partition	Used from OS partition
Configuration files	OS partition	OS partition	Stays in OS partition	Stays in OS partition	Used from OS partition	Used from OS partition
Online apps metadata	OS partition	OS partition	Stays in OS partition	Stays in OS partition	Used from OS partition	Used from OS partition
Maps and voice navigation	User partition	User partition	Deleted	Deleted	Downloaded from the Internet	Downloaded from the Internet

To reclaim storage for users, the OS performs data cleanup in two stages:

- The OS performs post-variant cleanup in some amount of time (default of 0 hours) after applying a variant for the user's primary SIM card and after completing initial device setup. Variant data is deleted from the user store because the device has been effectively branded during this time.

- The OS deletes all variant data from the user store in some amount of time (default of 72 hours) after completing initial device setup, if no variant has been applied to the device. No data type will be persisted on the device.

**Constraints:** ImageTimeOnly

**Instructions:**

- Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="MVDataManagement"
    Description="Use to configure various cleanup settings for runtime configuration
    data."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Multivariant">

            <!-- Set to 1 (to enable) or 0 (to disable) the backup of app installers for the selected
            variant when the device is branded.
            <Setting Name="PersistVariantData" Value="" />
            -->

            <!-- Set the time, in minutes, to wait after branding the device before deleting unused variant
            data from the user store.
            Maximum is 10080 or 7 days
            <Setting Name="PostVariantCleanupDelay" Value="" />
            -->

            <!-- Set the time, in minutes, to wait after finishing initial device setup before deleting all
            variant data from the user store.
            Maximum is 10080 or 7 days
            <Setting Name="UnconditionalCleanupDelay" Value="" />
            -->

        </Settings>

    </Static>

</ImageCustomizations>
```

- Specify an `Owner`.

- See the following sections for more information about the settings and the values you can set for each.

## Persist variant data

Use the `PersistVariantData` setting to configure runtime configuration to back up the app installers for the selected variant when the device is branded. The setting can be set to one of the following values:

VALUE	DESCRIPTION
0	Disable backup.
1	Enable backup. There must be sufficient space for runtime configuration backup to enable backup.

OEMs can configure the amount of reserved space to enable runtime configuration backup. To do this, set the

**MainOSRTCDATAReservedSectors** element in the OEMDevicePlatform.xml file.

**NOTE**

OEMs should only configure **MainOSRTCDATAReservedSectors** when using the runtime configuration feature to dynamically install certain applications from the Data partition depending on the SIM card(s) in the device during runtime. When using this functionality, the value is used to reserve space on the System partition to back up these applications so that they can be installed after a device reset.

When specifying the size, OEMs must specify a number of sectors that is sufficient to contain the latest get of applications placed in the data store that might be installed for an individual mobile operator. For example, if the OEM's customization answer file specified applications A, B, and C should be on the data partition and should only be installed for mobile operator Contoso, then the size reserved must be the size of A+B+C in MB and divided by 512 bytes per sector. At a maximum, OEMs can use **MainOSRTCDATAReservedSectors** to reserve sectors up to 100 MB to be used by the runtime configuration engine.

The following example shows how to reserve 50 MB:

```
<?xml version="1.0" encoding="utf-8"?>
<OEMDevicePlatform xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schemas.microsoft.com/embedded/2004/10/IMageUpdate">
  <MinSectorCount>20971520</MinSectorCount>
  <MainOSRTCDATAReservedSectors>102400</MainOSRTCDATAReservedSectors>
  <DevicePlatformID>{9D29F434-49E8-4C09-97AB-EF1DECC85D85}</DevicePlatformID>
</OEMDevicePlatform>
```

## Post variant cleanup delay

Use the **PostVariantCleanupDelay** setting to specify the time, in minutes, for the OS to wait after branding the device before deleting unused variant data from the user store. You can set this setting between 0 and 10080 minutes (or 7 days). If you specify a hexadecimal value, add the 0x prefix.

## Unconditional cleanup delay

Use the **UnconditionalCleanupDelay** setting to specify the time, in minutes, for the OS to wait after initial device setup is finished before deleting unused variant data from the user store. You can set this setting between 0 and 10080 minutes (or 7 days). If you specify a hexadecimal value, add the 0x prefix.

## Factory mode

This setting is not exposed through MCSF. OEMs can set the **Enable** value (REG\_DWORD) under the HKEY\_LOCAL\_MACHINE\Software\OEM\FactoryMode registry key to 1 (indicates factory mode) or 0 (not in factory mode). A dialer plugin or other mechanism used during factory testing can turn on factory mode to prevent runtime configuration backup/restore/cleanup of variant data as well as retail mode offline content cleanup.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Override the default CountryTable.xml

10/2/2018 • 2 minutes to read • [Edit Online](#)

The mobile runtime configuration package includes a built-in XML file (CountryTable.xml) for mapping between GeIDs, iso3166 country/region codes, and MCCs. Windows uses this table to assist in validating 3-digit MNCs for specific countries/regions, which is done by including an MNC list for that country/region. Otherwise, the OS assumes that MNCs are valid if they are 2 digits.

OEMs can override the default country/region lookup table and instruct the runtime configuration engine to use an OEM-provided mapping table instead.

**Constraints:** ImageTimeOnly

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="OverrideDefaultCountryLookup"
    Description="Use to override the default country/region lookup table
(CountryTable.xml) with the OEM mapping table."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Multivariant">
            <Setting Name="OverrideDefaultCountryLookup" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set the value for `OverrideDefaultCountryLookup` to one of the following values:

VALUE	DESCRIPTION
0	Use the default country/region lookup table, CountryTable.xml, used by the OS.
1	Override the default country/region lookup table and use an OEM provided mapping table.

4. If the value for `OverrideDefaultCountryLookup` is set to 1, the OEM must include their custom countrytable.xml file in a package and place this file in the %systemdrive%\programs\commonfiles\adc\OEM directory.

OEMs who provide their own country/region lookup table must use the following format for the XML file:

```
<countrytable>
  <country mcc="202" iso3166="GR" GeoID="98"/>      <!-- Greece -->
  <country mcc="204" iso3166="NL" GeoID="176"/>      <!-- Netherlands -->
  <!-- And so on -->
  <country mcc="316" iso3166="US" GeoID="244">      <!-- United States-->
    <mnclist>
      <mnc id="010"/>
      <mnc id="011"/>
    </mnclist>
  </country>
  <!-- And so on -->
</countrytable>
```

## Testing:

Work with your mobile operator partners to test this customization on their networks.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Setting the UICC slot for branding configuration

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can specify which UICC slot will be used for branding configuration.

**Constraints:** ImageTimeOnly

**Instructions:**

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="SetBrandingSlot"
    Description="Use to specify which UICC slot to use for branding configuration."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Multivariant">
            <Setting Name="BrandingSlot" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.
3. Set the `BrandingSlot``Value` to use one of the following values depending on which UICC slot you want to use for branding configuration:

VALUE	DESCRIPTION
0	Slot 0
1	Slot 1

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Customizations for hardware components

10/2/2018 • 3 minutes to read • [Edit Online](#)

This section contains information about customization settings that OEMs can use for the following hardware components:

- Buttons
- Camera
- Display
- Networking
- Sensors
- Storage
- Touch

## In this section

TOPIC	DESCRIPTION
<a href="#">Buttons: Enabling the Start button to wake the phone</a>	OEMs can configure the Start button to wake up the phone from the sleep state (also sometimes called the idle state). This can be configured on phones with one of the following hardware configurations: <ul style="list-style-type: none"><li>• The phone has a hardware Start button.</li><li>• The phone uses capacitive buttons, and the buttons share the same touch controller as the display panel but use separate sense lines, or the buttons have a dedicated touch controller.</li></ul>
<a href="#">Camera: Improved user experience for phones without a HW camera button</a>	
<a href="#">Display: Building images for FWVGA panels with static software buttons</a>	OEMs can build images for FWVGA (480 x 854) display panels where the Back, Home, and Search buttons are rendered on the screen by the OS instead of using hardware buttons. In these types of images, the bottom 54 scan lines of the display panel are reserved to render the software buttons.
<a href="#">Display: Building images with user-managed software buttons</a>	
<a href="#">Networking: Configuring the MTU data size</a>	For TCP, the default maximum transmission unit (MTU) is set to 1500 bytes, and the maximum segment size (MSS) is 1460 bytes. In general, this value should not be changed, as the user experience will degrade if low values are set. However, if the MSS does not meet the requirements of the mobile operator network, OEMs can customize it by setting the MTU data size.

Topic	Description
Sensors: Auto brightness	<p>This customization allows partners to customize the brightness by specifying:</p> <ul style="list-style-type: none"> <li>The value of brightness when dimming the screen.</li> <li>The ABS millilux range mapping.</li> <li>The ABS intensity percent mapping.</li> <li>The brightness state transition delay (in seconds).</li> </ul>
Storage: Enabling the packed commands feature for eMMC	<p>You can configure the phone to enable the packed commands feature for eMMC parts. This feature packs multiple small read requests in a single transfer request to the eMMC device.</p>
Storage: Enabling the UHS-1 feature for SD cards	<p>You can configure the phone to enable the UHS-1 feature for SD cards. This feature enables a higher bus speed for SD cards that support Ultra High Speed Phase I (UHS-1).</p>
Storage: Enabling the HS200 feature for eMMC	<p>You can configure the device to enable the HS200 feature for eMMC parts. This eMMC feature delivers a theoretical throughput of 200 MB/s across the eMMC bus, using a 200 MHz single data rate clock with an 8-bit bus width. This can result in significant performance improvements, especially on higher-end eMMC parts.</p>
Touch: Defining capacitive button behavior	<p>For mobile devices that use capacitive buttons, OEMs must add registry values that specify the number of capacitive buttons, the button locations, the button names, and the values to send to the OS when the button is pressed. This information enables the OS to treat the capacitive buttons below the screen as touchable targets.</p>
Touch: Describing the physical width and height of the display	<p>As part of implementing support for the touch controller hardware, OEMs must add registry values that specify the physical width and height the portion of the screen that is used to render the mobile device UI. The OS uses this information to properly scale touch gestures and help ensure a fluid user experience.</p>
Touch: Specifying the repeat rate for touch samples during touch-and-hold presses	<p>As part of implementing the touch driver, OEMs must determine how to send repeated touch samples to the input reader component in the OS during touch-and-hold presses. OEMs can choose to have the HID touch class driver (TchHID.sys) automatically send duplicate data packets to the input reader component in the OS during touch-and-hold presses, or they can send repeated touch samples from their touch driver. The OEM must add a registry value that tells the OS which implementation they chose, and the repeat rate to use if the OEM chose to have TchHID.sys send duplicate data packets automatically.</p>

Topic	Description
<a href="#">Wi-Fi: Removing cellular functionality on the mobile device</a>	If your mobile device does not support a cellular radio or will not be connected to a cellular network, you can remove all cellular-related functionality from the device's user interface by adding the WIFI FEATURE PACK feature entry in your OEMInput.xml file.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Enabling the Start button to wake the phone

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can configure the Start button to wake up the phone from the sleep state (also sometimes called the idle state). This can be configured on phones with one of the following hardware configurations:

- The phone has a hardware Start button.
- The phone uses capacitive buttons, and the buttons share the same touch controller as the display panel but use separate sense lines, or the buttons have a dedicated touch controller.

## Note

Although OEMs typically configure this behavior by adding a registry value in an INF file that is included in a driver package, this behavior can also be configured via the customization process described below. By using both options, OEMs can define the default behavior in the driver for a specific hardware component and modify this behavior as necessary in images for different phone models that use the same driver.

**Constraints:** None

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>

<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="StartOnIdle"
    Description="Use to specify whether the Start button can wake the phone during the
idle state."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Input/Keyboard/EnabledOnIdleButtons">
            <Setting Name="EnableStartOnIdle" Value="" />
        </Settings>
    </Static>

</ImageCustomizations>
```

2. Specify an `Owner` value in the customization answer file.

3. For the `EnableStartOnIdle` setting, set the `Value` to one of the following values.

VALUE	DESCRIPTION
0	Prevent the Start button from waking the phone during the idle state. This is also the behavior if you do not define this setting.
1	Enable the Start button to wake the phone during the idle state.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Improved user experience for phones without a HW camera button

10/2/2018 • 2 minutes to read • [Edit Online](#)

On mobile devices that do not have a hardware camera button present, OEMs can provide a better user experience by enabling this customization so that parts of the UI and service are updated to indicate to the user that there is no camera button on the device. When this customization is enabled:

- The camera roll text will not include the string **camera button**.
- The **camera** settings screen will not include the string **camera button**.
- The **camera** settings screen will not include settings that are button-specific.

When the user launches the camera app, the device displays a message on the viewfinder that shows the user how to take a picture by tapping on the screen.

**Constraints:** ImageTimeOnly

**Instructions:**

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="HWCameraShutterButtonNotPresent"
    Description="Use to provide a better user experience for phones that do not have a
hardware camera button."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Photos/OEM">
            <Setting Name="HWCameraShutterButtonNotPresent" Value="" />
        </Settings>
    </Static>

</ImageCustomizations>
```

2. Specify an `Owner` value in the customization answer file.
3. Set the value for `HWCameraShutterButtonNotPresent` to one of the following values:

VALUE	DESCRIPTION
0 or 'False, present'	Indicates that there is a HW camera shutter button. This is the default value.
1 or 'True, not present'	Indicates that there is no HW camera shutter button.

**Testing:**

1. Flash the build that contains this customization to a mobile device without a dedicated hardware camera

button.

2. Navigate to the camera **Settings** screen and verify that **camera button** is not displayed in the UI.
3. Verify that you do not see the following options in the UI:
  - **Press and hold camera button**

4. Navigate to the camera roll. If the camera roll is empty, verify that the caption displayed on the empty camera roll does not reference the camera button.

If the default camera app is configured to work above the lock screen, verify that the app that launches is the OEM lens app that you chose.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Building images for FWVGA panels with static software buttons

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can build images for FWVGA (480 x 854) display panels where the Back, Home, and Search buttons are rendered on the screen by the OS instead of using hardware buttons. In these types of images, the bottom 54 scan lines of the display panel are reserved to render the software buttons.

**Constraints:** None

## Instructions:

To build an FWVGA image with software-rendered buttons:

1. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="FWVGASoftwareButtons"
    Description="Use to generate an image that supports an FWVGA display panel and
software Back, Start, and Search buttons."
    Owner=""
    OwnerType="OEM">
<Static>

    <Settings Path="Graphics/D3D/DisplayResolutionOverride">
        <Setting Name="DisplayPanelOverrideWidth" Value="0x1E0" />
        <Setting Name="DisplayPanelOverrideHeight" Value="0x356" />
        <Setting Name="TouchPanelOverrideWidth" Value="0x1E0" />
        <Setting Name="TouchPanelOverrideHeight" Value="0x356" />
    </Settings>

    <Settings Path="Input/Touch/CapButtons">

        <Setting Name="ButtonCount" Value="" />
        <Setting Name="ButtonAreaTotal" Value="" />

        <Setting Name="Name0" Value="" />
        <Setting Name="VKey0" Value="" />
        <Setting Name="Area0" Value="" />

        <Setting Name="Name1" Value="" />
        <Setting Name="VKey1" Value="" />
        <Setting Name="Area1" Value="" />

        <Setting Name="Name2" Value="" />
        <Setting Name="VKey2" Value="" />
        <Setting Name="Area2" Value="" />

        <Setting Name="VibrateSupport" Value="" />
        <Setting Name="VibrateDuration" Value="" />
        <Setting Name="VibrateIntensity" Value="" />
    </Settings>

    <Settings Path="Shell/NavigationBar">
        <Setting Name="Color" Value="" />
    </Settings>
</Static>
</ImageCustomizations>

```

2. Specify an `Owner` value in the customization answer file.
3. Do not change the `DisplayPanelOverrideWidth`, `DisplayPanelOverrideHeight`, `TouchPanelOverrideWidth`, and `TouchPanelOverrideHeight` values shown in the example.
4. For the `ButtonCount`, `ButtonAreaTotal`, `Area n`, `Name n`, and `VKey n` settings, specify values as described in [Defining capacitive button behavior](#).
5. **Optional.** To enable the built-in vibration feedback mechanism for the buttons, include the `VibrateSupport`, `VibrateDuration`, and `VibrateIntensity` settings and specify values as described in [Defining capacitive button behavior](#). If you do not want to enable vibration feedback, you can omit these settings from the customization answer file.
6. **Optional.** To specify the default color of the software-rendered Back, Home, and Search buttons, include the `Color` setting and set `Value` to one of the following values. If you do not want to specify the default color, you can omit the `Color` setting and its parent `Settings` element.
  - `Black`
  - `Blue`
  - `Cyan`
  - `Magenta`
  - `Red`
  - `Yellow`

VALUE	DESCRIPTION
0	The buttons are black. This is the default value.
1	The color of the buttons matches the current theme chosen by the user (black for dark theme, white for light theme).
2	The color of the buttons matches the current accent color chosen by the user.

7. If you have not done so already, create an OEMInput XML file to define the set of packages to include in your phone image, and configure your computer to use the imaging tools. For more information, see [Building a mobile image using ImgGen.cmd](#).
8. In your OEMInput XML file, set the value of the **Resolution** element to 480x800. This configuration ensures that WVGA assets are used for the FWVGA image. Only WVGA assets can be used in FWVGA images that use software-rendered buttons.
9. In your OEMInput XML file, add the **NAVIGATIONBAR** feature to the **Microsoft** element that is a child of the **Features** element. This feature adds a phone setting that enables users to configure the color of the software buttons.

```
<Features>
  <Microsoft>
    <Feature>NAVIGATIONBAR</Feature>
  </Microsoft>
</Features>
```

For more information, see [Optional features for building images](#).

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Building images with user-managed software buttons

10/2/2018 • 7 minutes to read • [Edit Online](#)

On mobile devices with 1080p, 720p, WXGA, or qHD display panels without hardware Back, Start, and Search buttons, OEMs can build mobile images where the OS renders the Back, Start, and Search buttons in a navigation bar directly on the display. In this scenario, the buttons are not persistent; users can show or hide the buttons.

Optionally, OEMs can also customize the user experience for the navigation bar including:

- Changing the default color of the navigation bar background: light, dark, or a color that matches the device accent color.
- Enabling the navigation bar to be automatically hidden or shown.

The user-managed software buttons are optional, so OEMs must add the **NAVIGATIONBAR** feature in the OEMInput.xml file before the settings can be configured properly. A navigation bar Settings CPL is also added to enable users to change the default settings. In addition, OEMs must also set the correct value for the **Resolution** element.

OEMs should note the following accessibility features when support for the navigation bar is added:

- During initial phone setup, the navigation bar is always visible and cannot be dismissed.
- When the Narrator is active, the navigation bar is shown and cannot be dismissed until the Narrator is disabled.
- The Narrator can narrate the controls on the navigation bar.
- The buttons on the navigation bar provide haptic feedback.

## NOTE

Some settings in this customization may not be available depending on the Windows Phone 8.1 release that you are using to commercialize your device.

**Constraints:** ImageTimeOnly

**Instructions:** To build an image with software-rendered buttons that can be shown or hidden by users:

1. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="UserManagedSoftwareButtons"
    Description="Use to create an image that supports a 1080p, 720p, WXGA, or qHD
display
    panels without hardware Back, Start, and Search buttons."
    Owner=""
    OwnerType="OEM">

<Static>

<Settings Path="Input/Touch/CapButtons">

    <Setting Name="ButtonCount" Value="" />
    <Setting Name="ButtonAreaTotal" Value="" />

    <Setting Name="Name0" Value="" />
    <Setting Name="VKey0" Value="" />
    <Setting Name="Area0" Value="" />

    <Setting Name="Name1" Value="" />
    <Setting Name="VKey1" Value="" />
    <Setting Name="Area1" Value="" />

    <Setting Name="Name2" Value="" />
    <Setting Name="VKey2" Value="" />
    <Setting Name="Area2" Value="" />

    <Setting Name="VibrateSupport" Value="" />
    <Setting Name="VibrateDuration" Value="" />
    <Setting Name="VibrateIntensity" Value="" />

</Settings>

<Settings Path="Shell/NavigationBar">
    <Setting Name="Color" Value="" />
    <Setting Name="AutoHide" Value="" />
    <Setting Name="SwipeUpToHide" Value="" />

    <!-- These settings are available for Windows Phone 8.1 GDR1 or later versions of the OS.
    <Setting Name="BurnInProtectionMode" Value="" />
    <Setting Name="BurnInProtectionIdleTimerTimeout" Value="" />
    <Setting Name="BurnInProtectionMaskSwitchingPeriod" Value="" />
    <Setting Name="BurnInProtectionWhiteReplacementColor" Value="" />
    <Setting Name="BurnInProtectionBlackReplacementColor" Value="" />
    <Setting Name="UserEducationHintDisable" Value="" />
    <Setting Name="DoubleTapOff" Value="" />
    -->
</Settings>

</Static>

</ImageCustomizations>

```

2. Specify an `Owner` value in the customization answer file.
3. For the `ButtonCount`, `ButtonAreaTotal`, `Area n`, `Name n`, and `VKey n` settings, specify values as described in [Defining capacitive button behavior](#).
4. **Optional.** To enable the built-in vibration feedback mechanism for the buttons, include the `VibrateSupport`, `VibrateDuration`, and `VibrateIntensity` settings and specify values as described in [Defining capacitive button behavior](#). If you do not want to enable vibration feedback, you can omit these settings from the customization answer file.
5. **Optional.** To specify the default color of the software-rendered Back, Home, and Search buttons, include

the `Color` setting and set `Value` to one of the following values. If you do not want to specify the default color, you can omit the `Color` setting and its parent `Settings` element.

VALUE	DESCRIPTION
0 or <code>Always Dark</code>	The buttons are black. This is the default value.
1 or <code>Theme Color</code>	The color of the buttons matches the current theme chosen by the user (black for dark theme, white for light theme).
2 or <code>Accent Color</code>	The color of the buttons matches the current accent color chosen by the user.

6. **Optional.** To configure the navigation bar auto show and hide option, set `AutoHide` to one of the following values:

VALUE	DESCRIPTION
0 or <code>Disabled</code>	Disables the auto show and hide option for the navigation bar. This is the default OS value.
1 or <code>Enabled</code>	Enables the auto show and hide option for the navigation bar.

7. **Optional.** If you are using Windows Phone 8.1 GDR1 or later versions of the OS, you can also configure these settings:

- To set the OLED burn-in protection flag for the nav bar, set `BurnInProtectionMode` to one of the following values:

VALUE	DESCRIPTION
0 or <code>Disabled</code>	Disables the nav bar option for burn-in protection mode. This is the default OS value.
1 or <code>Enabled</code>	Enables the nav bar option for burn-in protection mode.

- To set the timeout, in seconds, for the OLED burn-in protection idle timer, set the value for `BurnInProtectionIdleTimeout`. The default value is 60 seconds. If you change the default value, the new value must be 1 second or more.
- To set the OLED burn-in protection mask switching period, in milliseconds, set the value for `BurnInProtectionMaskSwitchingPeriod`. The default value is 10,000 milliseconds (10 seconds). If you change the default value, the new value must be 1 millisecond or more.
- OEMs can also change the color that the nav bar uses during OLED burn-in protection. There are two settings: one for white UI elements like the button icon's color when the device is using a light theme and accent color mode, and the other is for black UI elements like the black background when the device is using a dark theme.
  - To set the white replacement color for the nav bar when in burn-in protection mode, set the value for `BurnInProtectionWhiteReplacementColor` to the ARGB color that you want to use. The color must be specified in the format `0xFFxxxxxx`. The default value for the white replacement color is 0xFFA9A9A9.

- To set the black replacement color for the nav bar when in burn-in protection mode, set the value for `BurnInProtectionBlackReplacementColor` to the ARGB color that you want to use. The color must be specified in the format **0xFFxxxxxx**. The default value for the black replacement color is 0xFF323232.
- By default, the nav bar shows a user education hint, which is a small piece of UI that the user is required to interact with at least once to learn the swipe up gesture. Once the user has seen the hint, they can choose to stop showing the hint so that it disappears the next time the user swipes up from the bottom of the screen to show the nav bar.

For OEMs building test and health images, you can configure the user hint by setting `UserEducationHintDisable` to one of the following values:

VALUE	DESCRIPTION
0 or <code>Hint Disabled</code>	Disables the user education hint.
1 or <code>Hint Enabled</code>	Enables the user education hint. This is the default OS behavior.

- To allow users to double tap any open space on the nav bar to turn off the screen, configure the `DoubleTapOff` setting to one of the following values:

VALUE	DESCRIPTION
0 or <code>Disabled</code>	Disables the feature. This is the default OS behavior.
1 or <code>Enabled</code>	Enables the feature.

8. If you have not done so already, create an OEMInput XML file to define the set of packages to include in your phone image, and configure your computer to use the imaging tools. For more information, see [Building a mobile image using ImgGen.cmd](#).

9. In your OEMInput XML file, set the value of the **Resolution** element to 1080x1920, 768x1280, 720x1280, or 540x960, depending on the resolution of the panel. For more information, see [OEMInput file contents](#).

In the following example, the highlighted entry shows what you need to add to the OEMInput XML file if you are supporting a 1080x1920 resolution.

```
<Resolutions>
    <Resolution>1080x1920</Resolution>
</Resolutions>
```

10. In your OEMInput XML file, add the **NAVIGATIONBAR** feature to the **Microsoft** element that is a child of the **Features** element. This feature adds a phone setting that enables users to configure the color of the software buttons.

```
<Features>
    <Microsoft>
        <Feature>NAVIGATIONBAR</Feature>
    </Microsoft>
</Features>
```

For more information, see [Optional features for building images](#).

### **Testing steps:**

The following scenarios are examples for testing the navigation bar and verifying the expected user experience based on your settings. You can add more testing scenarios including playing a game, capturing a screenshot, searching for content, browsing the Internet, and so on.

1. Flash a build containing this customization to a mobile device that does not have dedicated capacitive navigation buttons.
2. Go through the initial device setup process and verify whether the navigation bar can be hidden using a swipe gesture or by tapping the chevron (or ^ symbol facing downwards). Tap the Start button and verify that the navigation panel can be hidden or shown using a gesture or the chevron.
3. When the navigation bar is up, play a video and verify that the software buttons are automatically hidden with a transition. Tap on the screen to show the controls and navigation bar. Forward the video and then let it play for a while until the navigation bar auto hides itself. Show the navigation bar again by tapping on the screen and tap the back button to close the video.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Configuring the MTU data size

10/2/2018 • 2 minutes to read • [Edit Online](#)

For TCP, the default maximum transmission unit (MTU) is set to 1500 bytes, and the maximum segment size (MSS) is 1460 bytes. In general, this value should not be changed, as the user experience will degrade if low values are set. However, if the MSS does not meet the requirements of the mobile operator network, OEMs can customize it by setting the MTU data size.

**Note** This customization configures the MTU so the size should be set to the required MSS size plus 40 bytes.

**Constraints:** ImageTimeOnly

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="MTUDataSizeSettings"
    Description="Use to configure the MTU data size or roaming MTU data size."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="CellCore/PerDevice/External/ImageOnly">
            <Setting Name="MTU/MTUDataSize" Value="" />
            <Setting Name="MTU/RoamingMTUDataSize" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner` value in the customization answer file.
3. To change the default MTU data size, set the value for the `MTU/MTUDataSize` setting.
4. To change the default roaming MTU data size, set the value for the `MTU/RoamingMTUDataSize` setting.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Auto brightness

10/2/2018 • 3 minutes to read • [Edit Online](#)

The brightness of a mobile device display changes depending on the level of ambient light. For example, if you are using the device in a darkened movie theater, then walk into the medium light of the theater lobby, and then walk outside into bright sunshine, the brightness adjusts to the different light levels. The transition can be smooth or coarse depending on the capabilities of the brightness hardware. You can specify the light levels at which the brightness changes, the brightness for each of the levels, and the length of the transition period.

By design, the brightness is initially set when the brightness transitions from the OFF to ON state. From this initial setting, the brightness can be increased depending on the level of ambient light, but cannot be decreased. This helps to avoid the perceived flickering on the device as it moves through regions of both high and low ambient light levels. Brightness is reset at the next brightness OFF to ON transition.

The adaptive brightness service monitors ambient light levels by reading the ambient light sensor (ALS). Based on the user's configuration for manual or automatic brightness settings, adaptive brightness adjusts the display brightness as needed. Adaptive brightness is disabled when the service is idle, by reaching maximum brightness, or by being in manual brightness mode.

The device also monitors the user's activity from regular input to the device. If that activity stops in a period of time, the device will warn the user that the screen is about to turn off. It does so by changing the brightness of the screen to the value of `DimBrightness` a few seconds ahead of time. If the user touches any button or the screen then the screen brightness is restored and active monitoring resets. If no input is received, the device turns off. The time of activity monitoring can be configured in the **Screen times out after** option in the lock screen settings screen.

This customization allows partners to customize the brightness by specifying:

- The value of brightness when dimming the screen.
- The ABS millilux range mapping.
- The ABS intensity percent mapping.
- The brightness state transition delay (in seconds).

The value for `ABSRageMilliLuxMapping` is a list of values, separated by semicolons, that represent the upper bound in the range of ambient light readings measured by the light sensor. The upper bound value is measured in millilux. For example, if `ABSRageMilliLuxMapping` is set to 100000;500000;MAX, then the three brightness levels for the phone will be 100000 millilux, 500000 millilux, and maximum brightness. Specifying "MAX" in the list of values means that there is no upper bound to the range.

The ABS millilux range mapping and ABS intensity percent mapping are used together to make a table of brightness values. They must have the same number of elements as each other. For example if `ABSRageMilliLuxMapping` = 100000;500000;MAX and `ABSPercentIntensityMapping` = 33;66;100, the table will look like the following.

RANGE OF AMBIENT LIGHT READINGS IN MILLILUX	0-100000	100001-500000	500001 OR GREATER
Brightness	33%	66%	100%

**Constraints:** ImageTimeOnly

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="AutoBrightness"
    Description="Use to customize how the device screen adapts to brightness."
    Owner=""
    OwnerType="OEM">

<Static>
    <Settings Path="AutoBrightness">
        <!-- The value of brightness used when dimming the screen. The value must be an integer between 0 and 50 inclusive. -->
        <Setting Name="DimBrightness" Value="" />

        <!-- The ABS millilux range mapping. Set the value to a range, for example: "100000;500000;MAX". -->
        <Setting Name="ABSRangeMilliLuxMapping" Value="" />

        <!-- The ABS intensity percent mapping. Sample value: "11;68;100". -->
        <Setting Name="ABSPercentIntensityMapping" Value="" />

        <!-- The brightness state transition delay in seconds. The value must not be less than 2 seconds. Sample value: "5". -->
        <Setting Name="TransitionDelay" Value="" />
    </Settings>
</Static>
</ImageCustomizations>
```

2. Specify an `Owner` value in the customization answer file.

3. Set the `Value` for the following settings:

SETTING NAME	DESCRIPTION
<code>DimBrightness</code>	The dim brightness, in percent. The default value is 10.
<code>ABSRangeMilliLuxMapping</code>	The ALS mapping values in millilux.
<code>ABSPercentIntensityMapping</code>	The ALS mapping values in percent.
<code>TransitionDelay</code>	Delay, in seconds, from when the ambient light changes until the brightness begins its transition. This value has to be greater than or equal to 2. The default value is 5.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Enabling the packed commands feature for eMMC

10/2/2018 • 2 minutes to read • [Edit Online](#)

You can configure the phone to enable the packed commands feature for eMMC parts. This feature packs multiple small read requests in a single transfer request to the eMMC device.

**Constraints:** ImageTimeOnly

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>

<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="PackedCommands"
    Description="Use to enable the packed commands eMMC feature."
    Owner=""
    OwnerType="OEM">

    <Static>
        <Settings Path="Storage/SdBus/MainOS">
            <Setting Name="PackedCommandEnable" Value="" />
        </Settings>
    </Static>

</ImageCustomizations>
```

2. Specify an `Owner` value in the customization answer file.

3. Set the `Value` to one of the following:

VALUE	DESCRIPTION
0	Disable the packed commands feature. This is the default value.
1	Enable the packed commands feature.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Enabling the UHS-1 feature for SD cards

10/2/2018 • 2 minutes to read • [Edit Online](#)

You can configure the phone to enable the UHS-1 feature for SD cards. This feature enables a higher bus speed for SD cards that support Ultra High Speed Phase I (UHS-1).

**Constraints:** ImageTimeOnly

**Instructions:**

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>

<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="UHS1"
    Description="Use to enable the UHS-1 SD feature."
    Owner=""
    OwnerType="OEM">

    <Static>
        <Settings Path="Storage/SdBus/MainOS">
            <Setting Name="DisableUhsSupport" Value="" />
        </Settings>
    </Static>

</ImageCustomizations>
```

2. Specify an `Owner` value in the customization answer file.

3. Set the `Value` to one of the following:

VALUE	DESCRIPTION
0	Enable the UHS-1 feature.
1	Disable the UHS-1 feature. This is the default value.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Enabling the HS200 feature for eMMC

10/2/2018 • 2 minutes to read • [Edit Online](#)

You can configure the device to enable the HS200 feature for eMMC parts. This eMMC feature delivers a theoretical throughput of 200 MB/s across the eMMC bus, using a 200 MHz single data rate clock with an 8-bit bus width. This can result in significant performance improvements, especially on higher-end eMMC parts.

**Constraints:** ImageTimeOnly

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>

<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="HS200"
    Description="Use to enable the HS200 eMMC feature."
    Owner=""
    OwnerType="OEM">

    <Static>
        <Settings Path="Storage/SdBus/MainOS">
            <Setting Name="DisableHS200Support" Value="" />
        </Settings>
    </Static>

</ImageCustomizations>
```

2. Specify an `Owner` value in the customization answer file.

3. Set the `Value` to one of the following:

VALUE	DESCRIPTION
0	Enable the HS200 feature.
1	Disable the HS200 feature. This is the default value.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Defining capacitive button behavior

10/2/2018 • 3 minutes to read • [Edit Online](#)

For mobile devices that use capacitive buttons, OEMs must add registry values that specify the number of capacitive buttons, the button locations, the button names, and the values to send to the OS when the button is pressed. This information enables the OS to treat the capacitive buttons below the screen as touchable targets.

## Note

Although OEMs typically configure this behavior by adding a registry value in an INF file that is included in a driver package, this behavior can also be configured via the customization process described below. By using both options, OEMs can define the default behavior in the driver for a specific hardware component, and modify this behavior as necessary in images for different device models that use the same driver.

**Constraints:** ImageTimeOnly

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="CapButtons"
    Description="Use to describe the capacitive button area, names, and behavior."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Input/Touch/CapButtons">

            <Setting Name="ButtonCount" Value="" />
            <Setting Name="ButtonAreaTotal" Value="" />

            <Setting Name="Name0" Value="" />
            <Setting Name="VKey0" Value="" />
            <Setting Name="Area0" Value="" />

            <Setting Name="Name1" Value="" />
            <Setting Name="VKey1" Value="" />
            <Setting Name="Area1" Value="" />

            <Setting Name="Name2" Value="" />
            <Setting Name="VKey2" Value="" />
            <Setting Name="Area2" Value="" />

            <Setting Name="VibrateSupport" Value="" />
            <Setting Name="VibrateDuration" Value="" />
            <Setting Name="VibrateIntensity" Value="" />

        </Settings>
    </Static>

</ImageCustomizations>
```

2. Specify an `Owner` value in the customization answer file.
3. For the `ButtonCount` setting, set the `Value` to the number of capacitive buttons. This must be a value less than or equal to 3. If the device has a combination of capacitive buttons and hardware buttons, this entry

should only specify the number of capacitive buttons.

4. For the `ButtonAreaTotal1` setting, set the `Value` to a string that defines the overall capacitive button area beyond the LCD. The string must have the format `ul.x,ul.y lr.x,lr.y`, where `ul` = upper left and `lr` = lower right. For example, `Value="0,1280 768,1390"`.
5. For each capacitive button, the customization answer file must include a set of `Area n`, `Name n`, and `vKey n` settings, where `n` starts with 0. Configure these settings as described below.
  - For the `Name n` setting, set the `Value` to a string that identifies the current button. The only allowed strings are "Back", "Start", or "Search".
  - For the `vKey n` setting, set the `Value` to the value that is sent to the input pipeline when the current button is pressed. This must be one of the following values.

BUTTON	VALUE
Back	0x1B
Start	0x71
Search	0x72

- For the `Area`\*n\* setting, set the `Value` to a string that marks the position of the current button. The string must have the format \*ul.x,ul.y lr.x,lr.y\*, where \*ul\* = upper left and \*lr\* = lower right. For example, `Value=" 0,1295 236,1390"`.

1. If you want to enable the built-in vibration feedback mechanism for the capacitive buttons in the OS, include the `vibrateSupport`, `VibrateDuration`, and `VibrateIntensity` settings and configure them as described below. If you do not want to enable vibration feedback, you can omit these settings from the customization answer file.
  - For the `vibrateSupport` setting, set `Value` to one of the following values.

VALUE	DESCRIPTION
0	Disable vibration feedback. This is the default value.
1	Enable vibration feedback.

- For the `VibrateDuration` setting, set `Value` to a hexadecimal value between 0 and 1000 in decimal (or 0x0 and 0x3E8 in hexadecimal) that specifies the duration for a vibration, in milliseconds. The following example sets this value to 100.

```
```
<Setting Name="VibrateDuration" Value="0x64" />
```

```

- For the `VibrateIntensity` setting, set `Value` to a value between 0 and 100 in decimal (or 0x0 and 0x64 in hexadecimal) that specifies the intensity of the vibration. The following example sets this value to 50.

```
```
<Setting Name="VibrateIntensity" Value="0x32" />
```

```

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Describing the physical width and height of the display

10/2/2018 • 2 minutes to read • [Edit Online](#)

As part of implementing support for the touch controller hardware, OEMs must add registry values that specify the physical width and height the portion of the screen that is used to render the mobile device UI. The OS uses this information to properly scale touch gestures and help ensure a fluid user experience.

## Note

Although OEMs typically configure this behavior by adding a registry value in an INF file that is included in a driver package, this behavior can also be configured via the customization process described below. By using both options, OEMs can define the default behavior in the driver for a specific hardware component, and modify this behavior as necessary in images for different device models that use the same driver.

**Constraints:** None

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>

<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="DisplayWidthAndHeight"
    Description="Use to specify the physical width and height the portion of the screen
that is used to render the phone UI."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Input/Touch/DisplayProperties">

            <!-- The following values are in 10's of micrometers. -->
            <Setting Name="DisplayHeight" Value="" />
            <Setting Name="DisplayWidth" Value="" />

        </Settings>
    </Static>

</ImageCustomizations>
```

2. Specify an `Owner` value in the customization answer file.
3. For the `DisplayHeight` setting, set the `Value` to the height of the display in 10's of micrometers, formatted as a hexadecimal value. For example, `Value="0x206C"` specifies a height of 83 mm.
4. For the `DisplayWidth` setting, set the `Value` to the width of the display in 10's of micrometers, formatted as a hexadecimal value. For example, `Value="0x1388"` specifies a width of 50 mm.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Specifying the repeat rate for touch samples during touch-and-hold presses

10/2/2018 • 2 minutes to read • [Edit Online](#)

As part of implementing the touch driver, OEMs must determine how to send repeated touch samples to the input reader component in the OS during touch-and-hold presses. OEMs can choose to have the HID touch class driver (TchHID.sys) automatically send duplicate data packets to the input reader component in the OS during touch-and-hold presses, or they can send repeated touch samples from their touch driver. The OEM must add a registry value that tells the OS which implementation they chose, and the repeat rate to use if the OEM chose to have TchHID.sys send duplicate data packets automatically.

Microsoft recommends that OEMs send duplicate touch samples for touch-and-hold presses from their touch driver rather than have TchHID.sys automatically send repeated data packets.

## Note

Although OEMs typically configure this behavior by adding a registry value in an INF file that is included in a driver package, this behavior can also be configured via the customization process described below. By using both options, OEMs can define the default behavior in the driver for a specific hardware component, and modify this behavior as necessary in images for different mobile device models that use the same driver.

**Constraints:** None

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>

<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="AutoRepeat"
    Description="Use to specify the rate at which the HID touch class driver
(TchHID.sys) sends duplicate data packets to the input reader component in the OS during touch-and-hold
presses."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Input/Touch/AutoRepeat">
            <Setting Name="RepeatInterval" Value="" />
        </Settings>
    </Static>

</ImageCustomizations>
```

2. Specify an `Owner` value in the customization answer file.

3. For the `RepeatInterval` setting, set the `Value` to one of the following values:

- If your touch driver sends repeated touch samples to TchHID.sys during touch-and-hold presses, set this to a value of 0.
- If instead you want TchHID.sys to automatically send duplicate touch samples to the input reader component during touch-and-hold presses, set this to a value equal to or greater than 1, where the value is the repeat interval for duplicate touch samples in milliseconds.

- If you do not set this value, TchHID.sys will automatically send duplicate touch samples to the input reader component at a 50 millisecond interval.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Customizations for applications and Microsoft components

10/2/2018 • 2 minutes to read • [Edit Online](#)

This section contains information about customizations related to apps and Microsoft components.

## In this section

TOPIC	DESCRIPTION
<a href="#">Active phone cover settings</a>	OEMs can create and register an active phone cover app, which allows partners to create a user experience with their mobile device cover accessories. This app must be preloaded on the phone as a Settings/CPL application.
<a href="#">Customize the SIM toolkit</a>	OEMs can change the display duration for certain SIM toolkit UI dialogs or messages if the default values do not meet the requirements of the mobile operator.
<a href="#">Enhanced apps experience for medium and large screens</a>	OEMs can use the <code>UserPreferenceWidth</code> setting to override the default behavior based on the screen size and specify the physical width of the device (instead of using the automatically calculated <code>HORZSIZE</code> value).
<a href="#">Include required Microsoft components to the image</a>	This customization provides information on how partners can include the required Microsoft components in the OS image.
<a href="#">Phone calls/SMS filter applications</a>	OEMs can build and register a phone call/SMS filter application, which helps reduce the number of unwanted phone calls and text messages that users receive.
<a href="#">Preload an app with a dependency</a>	To preinstall an app that has dependencies on other packages or components, you must ensure the dependencies are preinstalled first, before your app. If the dependent packages or components are not installed first, your app preinstall will fail.
<a href="#">Remove optional Microsoft components from the image</a>	This customization provides information on how partners can remove any of the optional Microsoft components.
<a href="#">Store live tile</a>	The Store tile, when medium-sized, becomes a live tile. It shows both the Microsoft Store logo and the name. The Microsoft Store live tile cycles through apps that the user will see in the Store and lets the user discover apps outside of the Store. Microsoft recommends that partners keep the default Store live tile behavior. However, partners may change the default behavior to turn off the Store live tile and to prevent the OS from using cellular data to update the Store live tile in the background.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Active phone cover settings

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can create and register an active phone cover app, which allows partners to create a user experience with their mobile device cover accessories. This app must be preloaded on the phone as a Settings/CPL application.

OEMs can then enable the app to be launched when the active phone cover is closed and specify the default setting for the lock screen's auto unlock setting, which determines if the lock screen is automatically lifted when the user opens the cover.

## Limitations and restrictions:

When the OS receives a notification that the cover state has been set to Closed:

- The OS locks the device.
- The OS uses the **AUMID** setting to launch the active phone cover app. The app is launched in the foreground, above the lock screen, and the app is rendered at the top of the z-order.

When the OS receives a notification that the cover state has been set to Opened:

- The OS terminates the active phone cover app and shows the default lock experience.
- If the user opens the cover and the **AutoUnlock** setting is set to 1, the OS automatically lifts the lock screen and tries to unlock the device. If the device does not have a password lock, the OS unlocks the device. Otherwise, the OS prompts the user for their password.

**Constraints:** None

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="SmartCover"
    Description="Use to preload and configure your active phone cover app."
    Owner=""
    OwnerType="OEM">

    <Static>
        <!-- Preload the phone cover app. Specify the source, license, and ProvXML files. -->
        <Applications>
            <Application Source=""
                License=""
                ProvXML="" />
        </Applications>

        <Settings Path="Shell/SmartCover">
            <Setting Name="AUMID" Value="" />
            <Setting Name="AutoUnlock" Value="" />
        </Settings>
    </Static>
</ImageCustomizations>
```

2. Specify an **Owner**.

3. To preload the active phone cover app, add an **Applications** parent element and add an **Application** child element to correspond to the active phone cover app that you are preloading. For the **Application**, specify the values for the following settings:

- **Source** – Set to the path and name of the app to preload.
- **License** – Set to the path and name of the app license file.
- **ProvXML** – Set to the path and name of the provisioning XML file that corresponds to the app.

4. To enable the app to be launched when the cover is closed, set the value of **AUMID** to your app's Application User Model ID (AUMID). To identify the AUMID, follow the information in this [Microsoft Web site](#). The value must be in the format similar to this example: *SmartCoverApp\_<PublisherID>!App*.

5. To specify the default setting for the lock screen's auto unlock setting and determine if the lock screen is automatically lifted when the user opens the active phone cover, set **AutoUnlock** to one of the following values:

VALUE	DESCRIPTION
0 or 'Disabled'	The lock screen is not lifted when the user opens the active phone cover.
1 or 'Enabled'	The lock screen is automatically lifted when the user opens the active phone cover.

#### Testing steps:

1. Flash the build containing this customization to a phone.
2. Set up the phone and then go to **Settings** screen. Verify that the active phone cover app appears in the Settings CPL.
3. With the active phone cover attached to the phone, close the cover. Verify that the active phone cover app is launched successfully once the cover is closed.
4. Depending on the value you specified for the **AutoUnlock** setting, verify whether the lock screen is automatically lifted when you open the active phone cover.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Customize the SIM toolkit

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can change the display duration for certain SIM toolkit UI dialogs or messages if the default values do not meet the requirements of the mobile operator.

The default display times for SIM toolkit commands are as follows:

- GET INPUT: 120 seconds
- DISPLAY TEXT: 60 seconds
- SELECT ITEM: 60 seconds
- GET INKEY: 60 seconds

OEMs can modify the values for the following settings.

SETTING	DESCRIPTION
<code>UIDefaultDuration</code>	<p>Specifies the default time, in milliseconds, that the DISPLAY TEXT, GET INKEY, PLAY TONE, or SELECT ITEM dialog should be displayed.</p> <p>The default value is 60000 milliseconds (60 seconds). The valid value range is 1-120000.</p>
<code>UIGetInputDuration</code>	<p>Specifies the default time, in milliseconds, that the GET INPUT dialog should be displayed.</p> <p>The default value is 120000 milliseconds (120 seconds). The valid value range is 1-120000.</p>

To customize these settings using MCSF, see the next section. If you're using Windows provisioning, use Windows Configuration Designer to customize the settings or write your own Windows provisioning answer file. Regardless of the framework that you use, you must determine if you need to use the per-device or per-IMSI setting.

**Constraints:** None

This customization supports: **per-IMSI** value, **per-device** value

## Instructions

1. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="SIMToolkitCustomization"
    Description="Use to modify certain SIM toolkit UI dialogs and messages."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>

        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <!-- Use for the per-IMSI case -->
        <Settings Path="CellCore/PerIMSI/$(__IMSI)/UTK">
            <Setting Name="UIDefaultDuration" Value="" />
            <Setting Name="UIGetInputDuration" Value="" />
        </Settings>

        <!-- Use for the per-device case -->
        <Settings Path="CellCore/PerDevice/UTK">
            <Setting Name="UIDefaultDuration" Value="" />
            <Setting Name="UIGetInputDuration" Value="" />
        </Settings>
    </Variant>
</ImageCustomizations>

```

2. Specify an `Owner`.
3. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
4. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
5. Set the values for `UIDefaultDuration` and `UIGetInputDuration`.

## Testing

1. Flash the build containing this customization to a phone.
2. Go to the **Cellular & SIM > Advanced options** settings screen to start the SIM toolkit UI app.
3. Verify that the duration that the UI dialogs and messages are displayed match the default values that you've set.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Enhanced apps experience for medium and large screens

10/2/2018 • 3 minutes to read • [Edit Online](#)

The mobile device screen size is determined by the OS based on the Extended Display Identification Data (EDID). The Start tile defaults to the following layouts based on the **HORZSIZE** value, in millimeters, of the **GetDeviceCaps** function:

HORZSIZE (MM)	SCREEN SIZE	START TILE LAYOUT
< 62	Small	4-column
62 ≤ HORZSIZE ≤ 74	Medium	6-column
≥ 74	Large	6-column

The following table shows the device categories and resolutions that are supported for medium and large screen sizes.

DISPLAY TYPE	RESOLUTION	ASPECT RATIO	DIAGONAL SIZE	MEDIUM SIZE SUPPORTED	LARGE SIZE SUPPORTED
1080p (FHD)	1080 x 1920	16:9	3.7" to 7"	Yes	Yes
720p (HD)	720 x 1280	16:9	3.7" to 7"	Yes	Yes
WXGA	768 x 1280	15:9	3.5" to 5"	No	No
WVGA	480 x 800	15:9	3.5" to 5"	No	No
FWVGA	480 x 854	16:9	3.5" to 5"	No	No

## Note

The screen width matrix shown above will not work for 15:9 panel definitions. WXGA and WVGA, which are 15:9 panels only, support the small screen size.

For devices that are automatically calculated to have medium and large screen sizes, the entire UI is scaled down unless the screen or an application has opted out of the behavior. For devices with medium screens, the UI is scaled 93%. For devices with large screens, the UI is scaled 78%. Across devices, in case of a fixed size UI element, these will appear to be the same physical size. If the item is full screen width, the item's width will increase with the size of the screen, but the height will maintain the same physical size. Other changes in the UI include the following:

- More text is displayed horizontally.
- More items are displayed on a vertical list.

- Spacing between UI elements, such as text or icons, is proportionally scaled down.
- Image sizes are proportionally scaled down.

OEMs can use the **UserPreferenceWidth** setting to override the default behavior based on the screen size and specify the physical width of the device (instead of using the automatically calculated **HORZSIZE** value). The value for **UserPreferenceWidth** influences the screen resolution scale factor. A reboot is required for the change to take effect on the chrome process or any apps that are already running. Note that this only has a meaningful impact on 720 x 1280 and 1080 x 1920 devices.

**Constraints:** None

#### Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="UserPreferenceWidth"
    Description="Use to override the default layout behavior based on the screen size
    and specify the physical width of the device to influence screen
    resolution scale factor."
    Owner=""
    OwnerType="OEM">

    <Static>
        <Settings Path="ScreenSize">
            <!-- Set the value in millimeters. Specify the value in decimal or hexadecimal (0x prefix) value.
            -->
            <Setting Name="UserPreferenceWidth" Value="" />
        </Settings>
    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.
3. Set the value of `UserPreferenceWidth` to the physical width of the device, in millimeters, to override the default screen layout. Use either the decimal or hexadecimal (with the 0x prefix) value.

#### Testing steps:

1. Flash the build that contains this customization to a phone.
2. Verify that the following changes have been enabled on the phone:

CHANGE	DESCRIPTION
System font size	Verify that the <b>Ringtone</b> label text in the <b>Settings</b> screen is smaller.
Email list view	Open an email inbox and verify that the email items are listed with smaller text and that there is a second preview line for each item.
Messaging chat bubbles	Open the <b>Messaging</b> app and open a conversation. Verify that the font size for the chat cards is smaller, which scales the size of the chat bubbles.

CHANGE	DESCRIPTION
System font size and list item size	Open the <b>People</b> hub and verify that the font size and icons for each contact are smaller.
Two-column to three-column grid view	Open <b>Photos</b> and go to the <b>Albums</b> screen. Verify that there are more columns and the tile sizes and spacing are smaller.
System font size and list item size	Open the Microsoft Store app and verify that the apps list contains smaller tiles and font size.
Two-column to three-column grid view	From the Microsoft Store app, navigate to either top music albums or new releases and verify that there are 3 columns and the tile sizes and spacing are smaller.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Include required Microsoft components to the image

10/2/2018 • 2 minutes to read • [Edit Online](#)

This customization provides information on how partners can include the required Microsoft components in the OS image. For more information about other features you can include or exclude from your image, see [Optional features for building images](#).

For a comprehensive list of required Microsoft components that must be included in a Windows 10 Mobile image, refer to the *OEM Policy Document (OPD) for Windows 10 Mobile*.

- For any phone or other device that is submitted for NAL certification in China (excluding Hong Kong SAR and Macau), partners must meet the following requirements:
  - Do not include the any Skype app, tile, or functionality as part of any Windows 10 Mobile image, including any test images and final images.
  - Install the MESSAGINGLITE package, which does not contain Skype, on all devices.
- For any devices other than those listed above for which MESSAGINGLITE is required, partners must install the MESSAGINGGLOBAL package, which contains Skype.

## Note

The OS selects the correct messaging package to include as part of the image based on different locale and language combinations and sets this as the default selection. OEMs don't need to select the correct messaging package to install, but should make sure that the correct package is chosen to meet the requirements.

## Instructions:

### To install the Messaging package that does not include Skype integration

1. Locate the OEMInput.xml file that you are using to define your image.
2. Find the **Features** section, and within the **Microsoft** child element, review the **Feature** elements.
3. Add a <Feature>MESSAGINGLITE</Feature> entry in your OEMInput.xml file.

```
<Features>
  <Microsoft>
    <Feature>MESSAGINGLITE</Feature>
  </Microsoft>
</Features>
```

4. Save the updated OEMInput.xml file and rebuild your OS image.

5. Verify that the Messaging app does not have Skype integration.

### To install the Messaging package that includes Skype integration

1. Locate the OEMInput.xml file that you are using to define your image.
2. Find the **Features** section, and within the **Microsoft** child element, review the **Feature** elements.
3. Add a <Feature>MESSAGINGGLOBAL</Feature> entry in your OEMInput.xml file.

```
<Features>
  <Microsoft>
    <Feature>MESSAGINGGLOBAL</Feature>
  </Microsoft>
</Features>
```

4. Save the updated OEMInput.xml file and rebuild your OS image.
5. Verify that the Messaging app includes Skype integration.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Phone call/SMS filter applications

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can build and register a phone call/SMS filter application, which helps reduce the number of unwanted phone calls and text messages that users receive. This application must be preloaded on the phone as a Settings/CPL application.

## Limitations and restrictions:

- The OEM application shall be a CPL/Settings application.
- The OEM application shall have a privacy policy that is presented to the user before the user enables filtering, and also is available from the application, for example, a *Privacy policy* or *About* link. Furthermore, the application shall not send any of the information from the user (including phone numbers, dates of communication, text messages, and so on) off the device or to any other application or service which might do so.
- The user shall explicitly enable filtering in order to begin filtering calls and/or SMS messages. Additionally, the application must provide a way for the user to disable filtering.
- The application shall indicate to the user that no MMS or IM messages will be blocked, and that the user may still be charged for blocked SMS or blocked phone calls.
- The OEM application shall show blocked calls when launched from the **blocked calls** screen and must show blocked messages when launched from the **blocked messages** screen. Additionally, the OEM application shall require the user to explicitly consent to block or unblock a number when launched from the **block number...** option.

**Constraints:** FirstVariationOnly

## Instructions:

1. Create a phone call/SMS filter application.
2. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="PhoneCallSMSFilterApp"
    Description="Use to preload and configure your phone call/SMS filter application."
    Owner=""
    OwnerType="OEM">

    <Static>
        <!-- Preload the phone call/SMS filter app. Specify the source, license, and ProvXML files. -->
        <Applications>
            <Application Source=""
                License=""
                ProvXML="" />
        </Applications>

        <Settings Path="Phone/PhoneSmsFilter">
            <!-- Specify the app ID or GUID for your phone call/SMS filter app.
            The format looks like {12345678-1234-1234-1234-123456781234} -->
            <Setting Name="AppId" Value="" />
        </Settings>
    </Static>

</ImageCustomizations>

```

3. Specify an **Owner**.

4. To preload the phone call/SMS filter application, add an **Applications** parent element and add an **Application** child element to correspond to the phone call/SMS filter app that you are preloading. For the **Application**, specify the **Source** (.xap), **License**, and **ProvXML** files that correspond to app.
5. Set the value of **AppId** to the app ID or GUID for your phone call/SMS filter app. The value must be in the format **{12345678-1234-1234-1234-123456781234}**.

#### Testing steps:

1. Flash the build containing this customization to a device. The phone call/SMS filter application will be pre-installed as a settings/CPL app.
2. After the device has finished initial setup, launch the OEM CPL app.
3. In the phone call/SMS filter app, verify the following:
  - The privacy policy shows up as expected and the user is requested to approve.
  - The user is prompted if they would like to begin filtering.
4. Depending on your implementation, verify that the app can do the following:
  - Recognize the phone numbers of blocked incoming calls.
  - Recognize the phone number and message from an incoming SMS.
  - Add or remove phone numbers from the blocked list.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Preload an app with a dependency

10/2/2018 • 5 minutes to read • [Edit Online](#)

OEMs can preload apps as long as they meet the requirements specified in the Windows 10 Mobile OEM Policy Document (OPD). For more information on how to create preloaded apps for mobile devices, see [Preinstallable apps for mobile devices](#).

If you need to preinstall an app that has dependencies on other packages or components, you need to make sure that the other packages or components are preinstalled first before your app. If the dependent packages or components are not installed first, your app preinstall will fail.

## Instructions

### Prerequisites:

1. Make sure you have all the files you need to preload the OEM app:
  - App source, such as an .appx or .appxbundle
  - License file - This should be included as part of the preinstallation package.
  - ProvXML file - See the section [Create a .provxml file for a preinstallable app](#) for information on how to do this. When specifying the value of the **ProductID** parameter, this value must match the GUID from the AppxManifest file in the preinstallation package.
2. Make sure you have the following files for the required component:
  - App source, such as .appx or .appxbundle
  - ProvXML file - This is a separate provisioning file from the one that corresponds to the primary app. When writing the provXML file, the value of the **ProductID** parameter must match the GUID from the AppxManifest file that corresponds to the required component.
3. Name your provXML files so that they meet the requirements outlined in this step.

To make sure the required component is installed first, name the provXML files associated with the component in such a way that the file names precede the provXML file name for the primary app you want to preinstall. The OS preinstall logic uses the provXML file names to determine the order that apps and components are preinstalled so naming your required component so it alphabetically precedes the primary preinstall app ensures the dependency is resolved.

For example, if you have a primary app called ContosoPartnerApp that's dependent on a framework called ContosoFramework, you can ensure the framework is installed first by using these naming suggestions:

- For the ContosoFramework provXML file, use a name that is similar to MPAP\_000ContosoFramework\_001.provxml.
- For the ContosoPartnerApp provXML file, use a name that is similar to MPAP\_aaaContosoPartnerApp\_001.provxml.

In the Windows file system, "000" takes precedence over "aaa" so naming your provXML files this way ensures that the Contoso\_FrameworkApp is installed before Contoso\_PartnerApp.

### To preload the apps, build, and flash the OS image:

1. Create an package that contains the source file, license file, and provisioning file for the primary app.
  - a. Write a .pkg.xml and specify the source file, license file, and the .provXML file that corresponds to the primary app.

The following code example shows how to do this.

```
<?xml version="1.0" encoding="utf-8"?>
<Package xmlns="urn:Microsoft.WindowsPhone/PackageSchema.v8.00"
    Owner=""
    OwnerType="OEM"
    ReleaseType="Production"
    Platform="PlatformName"
    Component="Phone"
    SubComponent="ContosoPartnerApp">
    <Components>
        <OSComponent>
            <Files>
                <File
                    Source="C:\Contoso\Customizations\Assets\ContosoPartnerApp.appxbundle"
                    DestinationDir="$(runtime.commonfiles)\Xaps" />
                <File
                    Source="C:\Contoso\Customizations\Assets\MPAP_aaaContosoPartnerApp_001.provxml"
                    DestinationDir="$(runtime.commonfiles)\Provisioning\OEM" />
                <File
                    Source="C:\Contoso\Customizations\Assets\ContosoPartnerApp_License.xml"
                    DestinationDir="$(runtime.commonfiles)\Xaps" />
            </Files>
        </OSComponent>
    </Components>
</Package>
```

1. Create an package that contains the source file, license file, and provisioning file for the primary app.
  - a. Write a .pkg.xml and specify the source file, license file, and the .provXML file that corresponds to the primary app.
2. Create an package that contains the source file for the required component and the provisioning file that corresponds to it.
  - a. Write a .pkg.xml and specify the source file and the .provXML file that corresponds to the required package or component.

The following code example shows how to do this.

```

<?xml version="1.0" encoding="utf-8"?>
<Package xmlns="urn:Microsoft.WindowsPhone/PackageSchema.v8.00"
    Owner=""
    OwnerType="OEM"
    ReleaseType="Production"
    Platform="PlatformName"
    Component="Phone"
    SubComponent="ContosoFramework">
<Components>
    <OSComponent>
        <Files>
            <File
                Source="C:\Contoso\Customizations\Assets\contoso-framework-app.appx"
                DestinationDir="$(runtime.commonfiles)\Xaps" />
            <File
                Source="C:\Contoso\Customizations\Assets\MPAP_000ContosoFramework_001.provxml"
                DestinationDir="$(runtime.commonfiles)\Provisioning\OEM" />
        </Files>
    </OSComponent>
</Components>
</Package>

```

- b. Specify the values for the **Owner**, **ReleaseType**, **Platform**, **Component**, and **SubComponent** elements.
  - c. Replace *C:\Contoso\Customizations\Assets\contoso-framework-app.appx* with the location and file name of the component's source file.
  - d. Replace *C:\Contoso\Customizations\Assets\MPAP\_000ContosoFramework\_001.provxml* with the location and file name of the package or component's provisioning file.
  - e. Save the .pkg.xml file.
  - f. Run PkgGen to generate the .spkg from the .pkg.xml.
3. Write down the location and names of the .spkg files that were generated for your primary app and the required component.
  4. Update your feature manifest file to include these new packages and define a feature name for them. For more information, see [Feature manifest file contents](#).
    - a. Edit the feature manifest file.
    - b. In the feature manifest file, locate the **OEM** group under **Features**.
    - c. Within the **OEM** group, define a feature name for the package containing the required component and the package containing the primary app.

The following code example shows how to do this.

```

<PackageFile Path="C:\Contoso\Customizations\Assets"
Name="Owner.Component.SubComponent.ContosoFramework.spkg" >
<FeatureIDs>
<FeatureID>CONTOSO_FRAMEWORK</FeatureID>
</FeatureIDs>
</PackageFile>

<PackageFile Path="C:\Contoso\Customizations\Assets"
Name="Owner.Component.SubComponent.ContosoPartnerApp.spkg" >
<FeatureIDs>
<FeatureID>CONTOSO_PARTNERAPP</FeatureID>
</FeatureIDs>
</PackageFile>

```

- d. For each **PackageFile** element, change the values of the **Path** and **Name** attributes to match the location and name of your framework and app .spkg files.
  - e. Provide a **FeatureID** for your framework and primary app. You'll use these IDs to include these features in your OEMInput.xml file.
  - f. Save your updated feature manifest file.
5. Update your OEMInput.xml file to include the new features that you defined in the previous step. For more information, see [OEMInput file contents](#).
- a. Edit your OEMInput.xml file.
  - b. In the OEMInput file, locate the **OEM** group under **Features**.
  - c. Within the **OEM** group, define a feature name for the required component primary app.

The following code example shows how to do this.

```

<OEM>
<Feature>CONTOSO_FRAMEWORK</Feature>
<Feature>CONTOSO_PARTNERAPP</Feature>
</OEM>

```

- d. Change the **Feature** entries to match the **FeatureIDs** for your required component and primary app.
  - e. Save your updated OEMInput file.
6. Use the OEMInput.xml file as one of the inputs to build the mobile image.

You can use ImgGen.cmd to build the image.

7. Depending on the mobile image type that you built, you may need to sign the image before you can flash it to the phone. For more information, see [Sign a full flash update \(FFU\) image](#).

## Testing

1. Flash the image that contains the preloaded app to a mobile device.
2. Set up the device.
3. Once the device is fully set up, go to the full apps list.
4. Verify that you can see your primary app listed with all the other apps in the device.

## Related topics

[Prepare for Windows mobile development](#)

## Customization answer file overview

# Remove optional Microsoft components from the image

10/2/2018 • 2 minutes to read • [Edit Online](#)

This customization provides information on how partners can remove any of the optional Microsoft components. For more information about other features you can include or exclude from your image, see [Optional features for building images](#).

For a comprehensive list of optional Microsoft components, refer to the *OEM Policy Document (OPD) for Windows 10 Mobile*.

- Windows 10 Mobile ships with the following optional Microsoft components: Skype Windows Phone Silverlight app, Facebook, MSN Sports, MSN Money, and Continuum.

An OEM or mobile operator partner may opt-out of installing these optional apps in their final image. Where there is a prepinned Start tile for the replaced optional component, partners must prepin a new Start tile in its place using one of the other Microsoft apps that OEMs can prepin to Start. Partners are encouraged to ship these components as part of their final OS image except in markets where these applications are not allowed.

- For any phone or other device that is intended to be sold in China, partners must not include any Facebook app or tile provided by Microsoft as part of any Windows 10 Mobile image, including any test images and final images.

## Instructions:

### To remove Skype Windows Phone Silverlight app from the OS image

1. Locate the OEMInput.xml file that you are using to define your image.
2. Find the **Features** section, and within the **Microsoft** child element, review the **Feature** elements.
3. If SKYPE is in the list of optional Microsoft features, delete the **Feature** entry from the list.

In the following example, the <FEATURE> entry shows what you need to delete from your OEMInput.xml file.

```
<Features>
  <Microsoft>
    <Feature>SKYPE</Feature>
  </Microsoft>
</Features>
```

4. Save the updated OEMInput.xml file and rebuild your OS image.
5. Verify that the Skype Windows Phone Silverlight app is no longer part of the OS image.

### To remove Facebook from the OS image and Start screen

1. Locate the OEMInput.xml file that you are using to define your image.
2. Find the **Features** section, and within the **Microsoft** child element, review the **Feature** elements.
3. If FACEBOOK is in the list of optional Microsoft features, delete the **Feature** entry from the list.

In the following example, the <FEATURE> entry shows what you need to delete from your OEMInput.xml

file.

```
<Features>
  <Microsoft>
    <Feature>FACEBOOK</Feature>
  </Microsoft>
</Features>
```

4. Save the updated OEMInput.xml file and rebuild your OS image.
5. Verify that Facebook is no longer part of the OS image and does not appear in the Start screen.

For partners that opt-out of Facebook, partners must prepin a new Start tile in its place using one of the other Microsoft components that OEMs can prepin to Start.

#### To remove Continuum from the OS image

1. Locate the OEMInput.xml file that you are using to define your image.
2. Find the **Features** section, and within the **Microsoft** child element, review the **Feature** elements.
3. If Docking is in the list of optional Microsoft features, delete the **Feature** entry from the list.

In the following example, the <FEATURE> entry shows what you need to delete from your OEMInput.xml file.

```
<Features>
  <Microsoft>
    <Feature>Docking</Feature>
  </Microsoft>
</Features>
```

4. Save the updated OEMInput.xml file and rebuild your OS image.
5. Verify that Continuum is no longer part of the OS image.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Store live tile

10/2/2018 • 2 minutes to read • [Edit Online](#)

The Store tile, when medium-sized, becomes a live tile. It shows both the Microsoft Store logo and the name. The Microsoft Store live tile cycles through apps that the user will see in the Store and lets the user discover apps outside of the Store. By default, the Store live tile is on and out-of-the-box, the live tile is only updated over Wi-Fi until the user enters the Store for the first time. After the user enters the Store, the OS will start using cellular data to update the Store live tile in the background.

Microsoft recommends that partners keep the default Store live tile behavior. However, partners may change the default behavior to turn off the Store live tile and to prevent the OS from using cellular data to update the Store live tile in the background.

Regardless of the default Store live tile settings, users have the option of changing the defaults by choosing the **Live Tile** settings in the Microsoft Store **Settings** screen.

**Constraints:** ImageTimeOnly

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample or use the sample `StoreLiveTile.xml` file.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="StoreLiveTile"
    Description="Use to configure the Store tile in the Start screen."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="StoreMoOemGroup">
            <Setting Name="OemMoCustomizedIsLiveTileEnabled" Value="" />
            <Setting Name="OemMoLiveTileOptInToCellularAfterStoreLaunch" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.
3. To set the default value for **Live Tile Show products on tile** option in the Microsoft Store **Settings** screen, set `OemMoCustomizedIsLiveTileEnabled` to one of the following values:

VALUE	DESCRIPTION
0 or 'Disabled'	Disables the live tile feature for the Store tile. The <b>Show products on tile</b> option is Off.

VALUE	DESCRIPTION
1 or 'Enabled'	<p>Enables the live tile feature for the Store tile. The <b>Show products on tile</b> option is On.</p> <p>This is the default OS value.</p>

4. To set the default value for **Live Tile Only update the tile when I'm on Wi-Fi** option in the Microsoft Store **Settings** screen, set `OemMoLiveTileOptInToCellularAfterStoreLaunch` to one of the following values:

VALUE	DESCRIPTION
0 or 'Disabled'	<p>Prevents the OS from using cellular data to update the Store live tile after the user enters the Store for the first time. The <b>Only update the tile when I'm on Wi-Fi</b> option is On.</p> <p>Although the feature does not use a lot of data, partners may want to disable live tile updates over cellular data to meet certain market requirements.</p>
1 or 'Enabled'	<p>Allows the OS to use cellular data to update the Store live tile in the background after the user enters the Store for the first time. The <b>Only update the tile when I'm on Wi-Fi</b> option is Off.</p> <p>This is the default OS value.</p>

#### Testing steps:

1. Flash a build containing this customization to a phone.
2. Verify that the Store live tile is medium-sized and pinned to the Start screen.
3. Go to the **Settings** screen in the Microsoft Store app, and check the default values for the following **Live Tile** options: **Show products on tile** and **Only update the tile when I'm on Wi-Fi**. Confirm that they match the default values that you set.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Customizations for boot, initial setup, and shutdown

10/2/2018 • 2 minutes to read • [Edit Online](#)

Use these customizations to configure the device boot, initial setup, or shutdown experience.

## In this section

TOPIC	DESCRIPTION
<a href="#">Configure the timezone confirmation page during setup</a>	Use to allow users to change the timezone and region during device setup.
<a href="#">Configuring a boot screen to display in the final boot screen slot</a>	By default, the Windows 10 Mobile logo is displayed as the final boot screen. However, partners can display a different screen for the final boot screen slot. The image must be in .JPG, .JPEG, or .PNG format.
<a href="#">Configuring boot battery charging behavior</a>	The boot (UEFI) environment contains a battery charging application (owned by Microsoft) that is responsible for charging the battery in pre-boot and low power states. OEMs can configure some of the behavior of this application by using the registry values described in this topic.
<a href="#">Configuring OEM and mobile operator boot screens</a>	Partners must add at least one, and no more than two, boot screens (also called <i>splash screens</i> ) that are displayed when the device is turned on. These screens are intended for partners to display branding elements or logos.
<a href="#">Configuring the duration of the first boot screen</a>	If partners specify two boot screens (in addition to the Windows 10 Mobile boot screen), they can modify the duration of the first boot screen. We recommend that partners choose a duration for the first boot screen so that the first and second boot screens appear for the same amount of time.
<a href="#">Custom shutdown screen</a>	Partners can add a static logo or background during shutdown.
<a href="#">Language selection during initial setup</a>	If multiple display languages are included on the device, partners have the option of hiding the <b>Language selection</b> screen during setup.

TOPIC	DESCRIPTION
<a href="#">Partner account configuration during setup</a>	<p>In Windows 10 Mobile, an OEM or mobile operator may specify one preloaded app to be launched at the end of setup to walk users through an OEM or mobile operator account setup.</p> <p> Optionally, an OEM or mobile operator may also preload an additional 4 apps that can be subrouted and called from a main app. In this case, the partner specifies one of the apps as the hub app (main app), which will be automatically launched at the end of setup. This app can then invoke other spoke apps (subrouted apps) to complete other tasks.</p>
<a href="#">Screen background color during initial setup</a>	<p>For Windows 10 Mobile, the default background during OOBE or initial device setup is always dark. To align with this change, OEMs can no longer change the default screen background color during OOBE or initial device setup.</p>
<a href="#">Set the default country/region when SIM PIN is on</a>	<p>OEMs can customize the default home country/region that shows up during OOBE in cases where the SIM PIN is turned on. This value is associated with the default ICCID values. When SIM PIN is turned off, the OS uses the MCC-derived country/region instead.</p>

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Configure the timezone confirmation page during setup

10/2/2018 • 2 minutes to read • [Edit Online](#)

Use to allow users to change the timezone and region during device setup.

By default, the OS shows the timezone confirmation page during initial device setup after the device receives the Network Identity and Time Zone (NITZ) information. This page allows users to change the timezone and region during setup. To meet requirements from some mobile operators, OEMs can hide this page.

**Constraints:** None

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="EnableNITZTimeDateConfirmation"
    Description="Use to configure the timezone confirmation page during initial device
setup."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="SetupWizard">
            <Setting Name="EnableNITZTimeDateConfirmation" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set the value of `EnableNITZTimeDateConfirmation` to one of the following values:

VALUE	DESCRIPTION
0 or 'Disable'	Hides the timezone confirmation page during setup.
1 or 'Enable'	Shows the timezone confirmation page during setup. This is the default OS value.

## Testing steps:

1. Flash a build containing this customization to a device capable of receiving NITZ information.
2. During the initial device setup process, verify that the timezone confirmation page is either hidden or shown in the UI depending on the value that you set for `EnableNITZTimeDateConfirmation`.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Configuring a boot screen to display in the final boot screen slot

10/2/2018 • 2 minutes to read • [Edit Online](#)

By default, the Windows 10 Mobile logo is displayed as the final boot screen. However, partners can display a different screen for the final boot screen slot. The image must be in .JPG, .JPEG, or .PNG format.

## **Limitations and restrictions:**

- If partners display a different boot screen for the final slot, the Windows 10 Mobile boot screen must be moved to the first or second slot by following the instructions in [Configuring OEM and mobile operator boot screens](#). When moving the Windows 10 Mobile boot screen to the first or second slot, partners must use one of the bitmaps available in the Windows 10 Mobile Kit under %WPDKCONTENTROOT%\WPBootScreens. The Windows 10 Mobile boot screen must not be removed, altered, or replaced.

## **Note**

The Windows 10 Mobile Kit provides a different bitmap for each of the supported screen resolutions: 480 × 800, 720 × 1280, 768 × 1280, and 1080 × 1920. OEMs should use the appropriate bitmap for the screen resolution supported by their hardware.

- The amount of time the final boot screen is displayed cannot be configured by the OEM. During cold boot, the final boot screen is displayed until device initialization is complete. During warm boot, the final boot screen is displayed for 2.5 seconds.
- Logos and images must be owned or licensed by the OEM or mobile operator partner.
- The final boot screen supports 24-bit-per-pixel bitmaps, and the screen cannot be animated.
- The bitmap for the final boot screen should match the screen resolution.
- Support for localization is not included. Images and text should be appropriate for the market in which the device will ship.
- Additional boot screens must not be added to the startup sequence.

**Constraints:** None

## **Instructions:**

1. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
Name="CustomBootScreen"
    Description="Use to display a different screen for the final boot screen slot."
    Owner=""
    OwnerType="OEM">

<Static>

<Settings Path="BootandShutdownScreens">
    <Asset Name="BootImage" Source="" />
    <Setting Name="WindowsPhoneBootScreenOverride" Value="" />
</Settings>

</Static>
</ImageCustomizations>

```

2. Specify an `Owner`.
3. Set the `Source` of the `BootImage` asset to the full path and name of the static logo or custom background you want to use when the device shuts down. For example, *C:\Program Files (x86)\Windows Kits\10\CustomizationAssets\CustomBootScreen\CustomLogo.png*.
4. Set the `WindowsPhoneBootScreenOverride` setting's value to the file name of the custom screen or logo that you just added. In the example, the value is *CustomLogo.png*.

## Related topics

[Configuring OEM and mobile operator boot screens](#)

[Configuring the duration of the first boot screen](#)

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Configuring boot battery charging behavior

10/2/2018 • 2 minutes to read • [Edit Online](#)

The boot (UEFI) environment contains a battery charging application (owned by Microsoft) that is responsible for charging the battery in pre-boot and low power states. OEMs can configure some of the behavior of this application by using the registry values described in this topic.

**Constraints:** ImageTimeOnly

## Instructions:

### Charging boot threshold

OEMs can specify the threshold at which the device boots from UEFI-based charging to the Main OS. It is necessary to hold the device in the UEFI charging phase to charge the battery when it is too low to boot into the Main OS.

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="ChargeSettings"
    Description="Use to specify the threshold at which the device boots from UEFI-based
    charging to the main OS and to enable power-off charging mode."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="ChargeSettings">
            <Setting Name="ChargingBootThreshold" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.
3. Set the `Value` to a threshold value between 1 and 10. Microsoft recommends that OEMs calibrate this value so that the device spends a minimal amount of time in the threshold charging mode.

#### Note

In Windows 10 Mobile, the UEFI charging app included in the OS is being deprecated. Microsoft recommends that OEMs use the app published by the SoC vendor. If you use the SoC vendor's app, make sure that you disable the Windows 10 app by updating ChargingBootThreshold to 255 (hex 0xFF).

### Power-off charging

This setting enables *power-off charging*. Power-off charging allows the device to charge while it appears off to the user. In power-off charging mode, the device does not boot to the OS when plugged in. Instead, the device waits for the user to press and hold the power button before booting to the OS.

#### Important

Power-off charging can only be configured when the device image is generated. Windows 10 Mobile does not provide a way for users to enable or disable power-off charging.

1. Create a customization answer file using the contents shown in the following code sample or use the sample ChargeSettings.xml file.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="ChargeSettings"
    Description="Use to specify the threshold at which the device boots from UEFI-based
charging to the main OS and to enable power-off charging mode."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="ChargeSettings">
            <Setting Name="PowerOffChargingEnabled" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.
3. Set the `Value` to 1 to enable power-off charging or to 0 to disable power-off charging.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Configuring OEM and mobile operator boot screens

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners must add at least one, and no more than two, boot screens (also called *splash screens*) that are displayed when the device is turned on. These screens are intended for partners to display branding elements or logos.

Boot screens are shown only when the device is booting. When the user turns on the device from a sleep state, only the lock screen appears. Boot screens or splash screens that were added are not applied until the next time that the device boots. The runtime configuration engine runs later in the boot sequence (after the splash screens have been shown) so the customization is not visible until the next time the device reboots.

## Limitations and restrictions:

- Logos and images must be owned or licensed by the OEM or mobile operator partner.
- The first two boot screens support 24-bit-per-pixel bitmaps, and they cannot be animated. For more information about the supported bitmap formats, see section 5.2.22.4 of the [ACPI Specification Revision 5.0](#).
- The bitmap, along with the (x,y) offset specified in the BGRT table, must fit inside the boundaries specified by the screen resolution.
- Support for localization is not included. Images and text should be appropriate for the market in which the device will ship.
- Additional boot screens must not be added to the startup sequence.
- If the OEM specifies two boot screens (in addition to the Windows 10 Mobile boot screen), the duration for displaying the first boot screen can be configured by the OEM. For more information, see [Configuring the duration of the first boot screen](#). The duration of the other boot screens cannot be altered.

## Instructions:

To specify only one boot screen:

- Update the BGRT table in ACPI to specify the required values to describe the boot screen. In particular, ensure that the **Image Address** field is set to the address of the bitmap to use for the boot screen image, and set the lowest bit of the **Status** field to 1. For more information about the BGRT table, see section 5.2.22 of the [ACPI Specification Revision 5.0](#). Because the implementation of the BGRT table is specific to the SoC vendor, the SoC vendor may have additional guidance or requirements for these changes.

To specify two boot screens:

1. Configure the image for the first boot screen to be loaded through UEFI. For more information, consult with the SoC vendor. In this scenario, the image for the first boot screen is not specified in the BGRT table; UEFI firmware alone is responsible for displaying the first boot screen.
2. For the second boot screen, update the BGRT table in ACPI to specify the required values to describe the second boot screen. In particular, ensure that the **Image Address** field is set to the address of the bitmap to use for the second boot screen image, and set the lowest bit of the **Status** field to 0 (the value 0 indicates that this BGRT table describes the second boot screen image). For more information about the BGRT table, see section 5.2.22 of the [ACPI Specification Revision 5.0](#). Because the implementation of the BGRT table is specific to the SoC vendor, the SoC vendor may have additional guidance or requirements for these changes.

## Related topics

[Configuring the duration of the first boot screen](#)

[Configuring a boot screen to display in the final boot screen slot](#)

# Configuring the duration of the first boot screen

10/2/2018 • 2 minutes to read • [Edit Online](#)

If partners specify two boot screens (in addition to the Windows 10 Mobile boot screen), they can modify the duration of the first boot screen. We recommend that partners choose a duration for the first boot screen so that the first and second boot screens appear for the same amount of time.

Only the duration of the first OEM-specified boot screen can be modified. The second OEM-specified boot screen appears when first boot screen duration is complete, and it is displayed until the boot process is complete. To determine the duration of the first boot screen, Microsoft recommends that OEMs time the boot process for their hardware, and choose a duration for the first boot screen that results in the first and second boot screens appearing for the same amount of time.

The following table describes the duration of each boot screen when three boot screens are used.

SCREEN	DURATION
First boot screen	<p>By default, this screen is displayed for at least 5 seconds, generally several seconds longer due to firmware and early boot initialization time as described in the following note. This duration is configurable by the OEM. Microsoft recommends that this screen be displayed for at least 2.5 seconds.</p> <div style="border: 1px solid black; padding: 5px;"><p><b>Note</b></p><p>The first boot screen typically appears for several seconds longer than the duration specified by the OEM, and the OEM should take this into consideration when determining a duration. The OEM-specified duration actually begins with the start of ntoskrnl.exe initialization, and typically the first boot screen actually appears several seconds before ntoskrnl.exe initialization starts.</p></div>
Second boot screen	<p>This screen is displayed from the time the first boot screen is finished until the boot process is complete and the primary display driver can display the third boot screen. This duration is not configurable by the OEM.</p>
Third boot screen (by default, the Windows Phone boot screen)	<p>During cold boot, this screen is displayed until phone initialization is complete. During warm boot, this screen is displayed for 2.5 seconds.</p>

For comparison, the following table shows the duration of each boot screen when only two boot screens are used.

SCREEN	DURATION
First boot screen	<p>This screen is displayed until the boot process is complete.</p>

SCREEN	DURATION
Second boot screen (by default, the Windows Phone boot screen)	During cold boot, this screen is displayed until phone initialization is complete. During warm boot, this screen is displayed for 2.5 seconds.

**Constraints:** ImageTimeOnly

**Instructions:**

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="FirstBootScreenDuration"
    Description="Use to modify the duration of the first boot screen if partners
specify an addition to the
Windows Phone boot screen."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="BootDisplaySettings">
            <Setting Name="BootUXLogoTransitionTime" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set the `BootUXLogoTransitionTime` value to the hexadecimal value of the number of milliseconds to display the first boot screen.

**Note**

Because the associated setting is a REG\_BINARY value, this value must be specified in hexadecimal pairs. For example, if you want to set the value to 1000 milliseconds or 0x03E8 (hexadecimal), you must set the value to "E8,03" in your customization answer file. Also note that the actual duration of the first screen may be several seconds longer than the value specified. For more information, see the note in the first table above.

## Related topics

[Configuring OEM and mobile operator boot screens](#)

[Configuring a boot screen to display in the final boot screen slot](#)

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Custom shutdown screen

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can add a static logo or background during shutdown. Shutdown screen images must be in .JPG, .JPEG, or .PNG format.

The resolution of the custom logo or background image provided by the partner should match the screen resolution of the device. If the image resolution is less than or greater than the screen resolution, the OS will scale the image to fit the screen.

## Limitations and restrictions:

- The **goodbye** text shall not be removed, altered, or replaced and the area behind the **goodbye** text must be dark enough for the text to remain legible.
- Any logos or images used in the shutdown screen shall be owned or licensed by the OEM or mobile operator partner.
- Partners shall not lengthen the amount of time the shutdown screen is displayed and thereby extend the amount of time it takes for the phone to shut down.

**Constraints:** None

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="CustomShutdownScreen"
    Description="Use to add a static logo during shutdown."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="BootandShutdownScreens">
            <Asset Name="ShutdownImage" Source="" />
            <Setting Name="WindowsPhoneShutdownScreenOverride" Value="" />
        </Settings>

    </Static>
</ImageCustomizations>
```

2. Specify an `Owner`.
3. Set the `Source` of the `ShutdownImage` asset to the full path and name of the static logo or custom background you want to use when the phone shuts down. For example, `C:\Program Files (x86)\Windows Kits\10\CustomizationAssets\CustomShutdownScreen\CustomLogo.png`.
4. Set the value of the `WindowsPhoneShutdownScreenOverride` setting to the file name of the custom screen or logo that you just added. For example, `CustomLogo.png`.

## Testing steps:

1. Flash a build containing this customization to a phone.

2. Go through initial phone setup.
3. Shut down the phone.
4. Verify that you can see the **goodbye** text superimposed over your custom static logo or background.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Language selection during initial setup

10/2/2018 • 2 minutes to read • [Edit Online](#)

If multiple display languages are included on the device, partners have the option of hiding the **Language selection** screen during setup. As a result, the device will use the default specified by the OEM, and users can change the language later by using the **Time & Language** screen in **Settings**.

**Constraints:** None

## Instructions:

1. Specify more than one phone language for your image. For more information, see [Phone languages](#).
2. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="LanguageSelectionScreen"
    Description="Use to show or hide the language selection screen during setup."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="SetupWizard">
            <Setting Name="ShowLanguageSelectionScreenInSetup" Value="" />
        </Settings>

    </Static>
</ImageCustomizations>
```

3. Specify an `Owner`.
4. Set the `Value` to either of the following:

VALUE	DESCRIPTION
0 or 'Hide'	Hides the <b>Language selection</b> screen during setup.
1 or 'Show'	Shows the <b>Language selection</b> screen during setup.

## Testing steps:

1. Flash a build containing this customization and multiple languages to a phone.
2. At the beginning of setup, verify that the **Language selection** screen is either shown or hidden depending on the default value you specified.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Partner account configuration during setup

10/2/2018 • 9 minutes to read • [Edit Online](#)

In Windows 10 Mobile, an OEM or mobile operator may specify one preloaded app to be launched at the end of setup to walk users through an OEM or mobile operator account setup.

Optionally, an OEM or mobile operator may also preload an additional 4 apps that can be subrouted and called from a main app. In this case, the partner specifies one of the apps as the hub app (main app), which will be automatically launched at the end of setup. This app can then invoke other spoke apps (subrouted apps) to complete other tasks.

Partners can use this customization to walk users through the process of setting up an OEM or mobile operator-specific account or to enable a multi-page OOBE setup experience. If partners configure this customization, the **All done** screen at the end of setup will be replaced with an **Almost done** screen that contains an introduction to the partner app.

## Limitations and restrictions

Partners must meet the following requirements when configuring this customization:

- An OEM or mobile operator can run up to five (5) apps at the end of OOBE on Windows mobile devices.

Only one partner, either the mobile operator or OEM, may have an app that launches at the end of OOBE.

- All the apps must be installed before the end of OOBE on the mobile device. This enables the hub app to invoke the spoke apps.

- All existing Windows Phone 8.1 and later customizations for launching an app at the end of OOBE remain in place. This includes the existing limitations and timing for unlocking home buttons.

○ The app must be preloaded and conform to all guidelines required of preloaded apps.

○ All text must be fully localized for the display languages that are included on the mobile device.

○ The app must provide a way for users to skip the task(s) and exit the app. For an example of a recommended UI for each account to configure, see the screen **Sign in with Microsoft account** that is shown during the standard device setup process.

○ The application can use and embed the browser control, but it cannot launch Microsoft Edge.

○ All pages and data necessary for the user to complete or skip the task must be included in the app. A data connection is not guaranteed, but an option to prompt the user to enable network connections is available. For more information, see the *Prompting the user to enable network connections* section below.

○ The app must look and function like a wizard, and provide buttons for the user to navigate forward and backward through the steps. The hardware keys—Back, Start, and Search—will not be available to the user.

○ The tasks in the app must not take longer than 5 minutes to complete. After that interval, the OS will resume control and run the final tasks to finish setup, including sending the Welcome SMS and turning on the hardware keys. If the app has not exited by that point, the user can press the Start hardware button to leave the app.

# Customization details

**Constraints:** None

## Instructions:

1. Create the partner account setup app that you want to launch at the end of OOBE.
2. Create a customization answer file to preload your app(s) and follow the steps in the section Configure the customization settings.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="PartnerAccountConfiguration"
    Description="Use to specify one preloaded app launched at the end of setup to walk
users
    through an OEM or mobile operator account setup. Partners may also
preload an
    additional 4 apps that may be subrouted and called from the main hub
app."
    Owner=""
    OwnerType="OEM">

<Static>

    <!-- Preload up to 5 apps to use for partner account setup. If you do, one app must be the hub app
        and the rest are spoke apps. -->
    <Applications>
        <Application Source="">
            License=""
            ProvXML="" />
        <Application Source="">
            License=""
            ProvXML="" />
    </Applications>

    <Settings Path="SetupWizard">
        <!-- Set to the GUID or App ID of the partner setup app -->
        <Setting Name="PartnerSetupAppID" Value="" />

        <!-- Specify the launch parameters for the partner setup app, such as the first page of the wizard
-->
        <Setting Name="PartnerSetupAppParameters" Value="/_default#/accountSetup.xaml" />

        <!-- Set to 1 to prompt the user to enable network connections before the app is run
-->
        <Setting Name="PartnerSetupAppNetworkPrompt" Value="1" />

        <!-- For one supported phone language, set the value to the localized partner name, account name,
and
            the name you want to appear in the Table of Contents. For multiple language support, use the
below
            settings instead as well as the Localization/MUI settings path for the base DLL and language
DLL files. -->
        <Setting Name="PartnerSetupAppPartnerName" Value="" />
        <Setting Name="PartnerSetupAppTaskName" Value="" />
        <Setting Name="PartnerSetupAppTOCTaskName" Value="" />
    </Settings>
</ImageCustomizations>
```

```

<!-- For multiple supported phone languages, use these settings and provide the base MUI DLL
<Setting Name="PartnerSetupAppPartnerName"
    Value="@c:\Data\Programs\{00000000-0000-0000-0000-
00000000000}\Install\DisplayNames.dll,-101" />
<Setting Name="PartnerSetupAppTaskName"
    Value="@c:\Data\Programs\{00000000-0000-0000-0000-
00000000000}\Install\DisplayNames.dll,-102" />
<Setting Name="PartnerSetupAppTOCTaskName"
    Value="@c:\Data\Programs\{00000000-0000-0000-0000-
00000000000}\Install\DisplayNames.dll,-103" />
-->

</Settings>

<!-- For multiple supported phone languages, add your base MUI DLL file and specify the
language MUI packages (*.dll.mui)

<Settings Path="Localization/MUI">
    <Asset Name="BaseDll" Source="" />

    <Asset Name="LanguageDll/${langid}" Source="" />
    <Asset Name="LanguageDll/${langid}" Source="" />
    <Asset Name="LanguageDll/${langid}" Source="" />
    <!-- Add as many as you need -->
</Settings>
-->

</Static>

</ImageCustomizations>

```

### Testing steps:

1. Flash the build containing this customization and relevant display languages to a mobile device.
2. Verify the **Almost done** screen displays the localized text as expected.
3. Launch and complete and partner setup app to verify it works appropriately, is localized, calls any other spoke apps, and that the user can exit instead of completing the setup tasks.

## App design considerations and guidelines

When designing your apps, keep the following design considerations in mind:

- We recommend that you use Universal Windows apps to enable your hub and spoke model.
- When building a shared partner app, be aware that the app might appear in the Microsoft Store for every mobile device produced by the OEM, every phone ranged by the mobile operator, and might be downloaded by users that have a different mobile operator. To ensure that users do not end up seeing mobile operator configuration options that do not apply to their phone or network, consider these possible mitigations:
  1. **Network verification:** The app must verify that the phone is on the specified mobile operator's network before displaying any account setup functionality for that mobile operator. This can be done by checking the MCC and the MNC of the SIM, or by verifying the registry value `PhoneMobileOperatorName` setting (see [Phone metadata in DeviceTargetingInfo](#) for more information).
  2. **System settings app:** The app can be written as a system settings app. This application will appear on the **System** screen in **Settings**, and cannot be pinned to Start. It also will be hidden in the Microsoft Store so that it cannot be accidentally downloaded.
- See [Design basics](#) to learn more about how you can design a Universal Windows app that suits a variety of

devices with different display sizes and other tips for creating an app with a great UI.

See [Sample app UI](#) for examples on what your partner setup app's UI might look like.

## MIDL bindings

MIDL bindings allow an app to be launched through a protocol and package family name. The hub and spoke model only works on Universal Windows apps and you must use [Windows.System.Launcher.LaunchUriForResultsAsync](#).

## Recommended APIs

We recommend that you use the following APIs when implementing your apps:

API	USECASE
<a href="#">Windows.System.Launcher.LaunchUriForResultsAsync</a>	<p>Allows a parent app to launch a child app. The child app must return to the parent app. This achieves the hub and spoke model requirement for this customization (for partners that want to use this model).</p> <p>When using this API, keep in mind the memory usage for mobile devices. Every time the API is used, another child app is launched while requesting not to terminate the parent app, which means that it must be kept in memory.</p>
<a href="#">Windows.System.Launcher.LaunchUriAsync</a>	Allows forward navigation from App 1 to App 2. The second app is not a child app of the first app.

## Configure the customization settings

### Preloading the apps and specifying the first page

1. Preload the apps to the mobile device using the following code example.

```
<!-- Preload up to 5 apps to use as the hub and spoke apps -->
<Applications>
    <Application Source="" 
        License="" 
        ProvXML="" />
    <Application Source="" 
        License="" 
        ProvXML="" />
</Applications>
```

2. In the MCSF customization answer file:

- If your app is a Universal Windows app, set `PartnerSetupAppID` to the AUMID for your app (or to your hub app if you have subrouted apps).  
If your app is not a Universal Windows app, set `PartnerSetupAppID` to the GUID or app ID for your app (or to your hub app if you have subrouted apps).
- Set `PartnerSetupAppID` to the GUID or app ID for your app (or to your hub app if you have subrouted apps), and set `PartnerSetupAppParameters` setting to the correct name of the first page of your hub app.

### Prompting the user to enable network connections

If the user selects a **Custom** configuration on the **Settings** screen in setup, and removes the checkmark from the **Allow cellular data usage on your phone** option, your app will not have network connectivity even if a valid

SIM is installed. To prompt the user to turn the network connection back on before the partner account configuration application is run, set the `PartnerSetupAppNetworkPrompt` setting to 1.

The values supported by this setting are:

VALUE	DESCRIPTION
1 or <code>Enable</code>	Prompt the user to enable network connections before the app is run.
0 or <code>Disable</code>	Do not prompt the user to enable network connections before the app is run.

### Localized strings for partner name, account name, and Table of Contents

Partners must provide three localized strings: partner name, name(s) of the account(s) to configure, and the name to show in the Table of Contents. The strings for the partner name and account name are used to complete the following statement displayed to the user in the screen before the app is launched.

"Almost done..."

"You're just about done setting up your phone. Next, `PartnerSetupAppPartnerName` will walk you through setting up `PartnerSetupAppTaskName`."

- If only one display language is included on the device, you can set the values for the `PartnerSetupAppPartnerName` and `PartnerSetupAppTaskName` settings directly to the desired strings.

You can also set the value for the `PartnerSetupAppTOCTaskName` setting directly. This string will be used for the Table of Contents.

In the above sample MCSF customization answer file, remove the comments in the section "For one supported phone language..." and set the values to the strings you want to use.

- If you include support for multiple display languages, you must create a resource-only .dll that contains these three strings localized into every included language.

#### NOTE

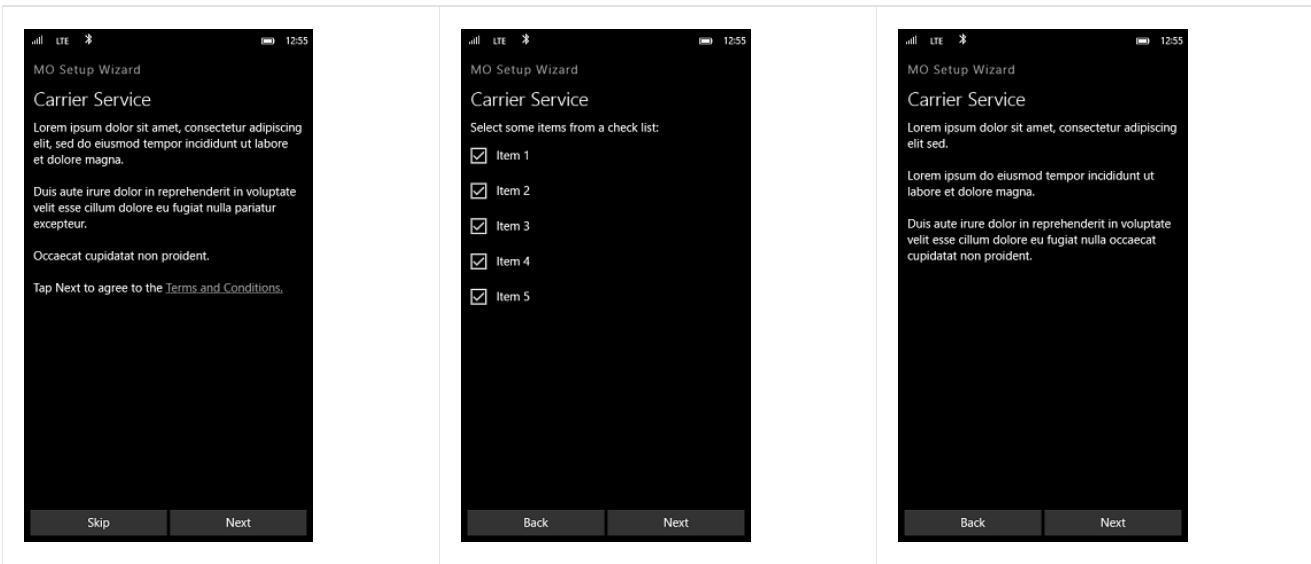
This capability is available only through the MCSF framework and will not work if you use the Windows provisioning framework.

In the above sample customization answer file, remove the comments around the section "For multiple supported phone languages...". The sample shows an installed .dll file named DisplayNames.dll, and that the partner name is string 101 and the account names are string 102. The string for the Table of Contents is string 103. Update these values to match your implementation and update the path to the path of your installed application.

## Sample app UI

The following screenshots show what a partner setup app's UI might look like and some of the tasks that the app may walk the user through during account setup.

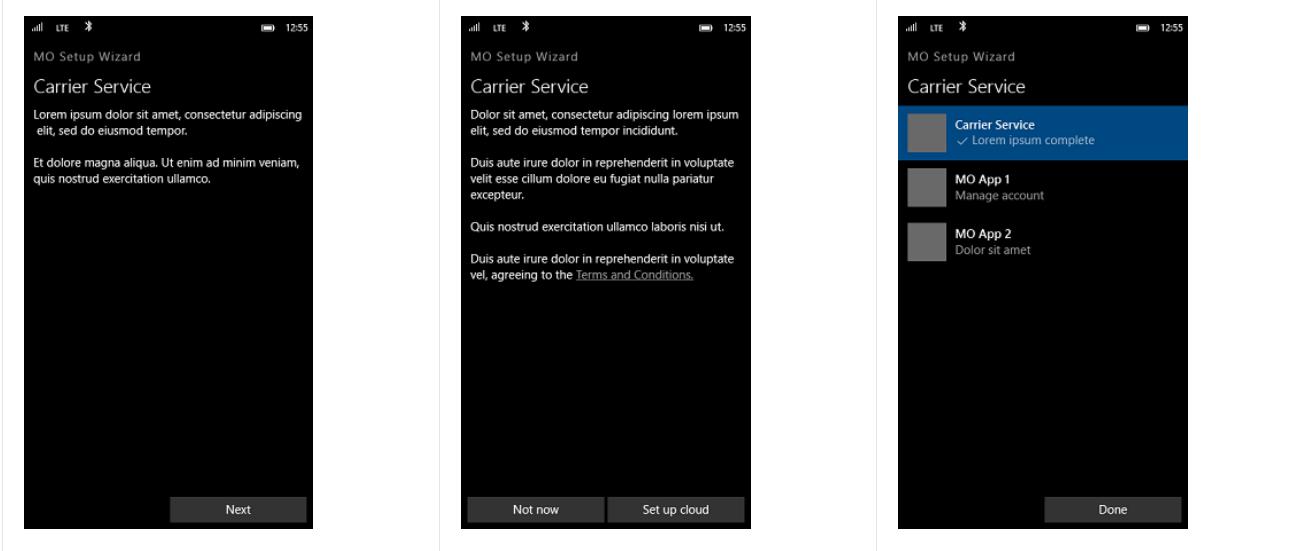
Screen 1	Screen 2	Screen 3
----------	----------	----------



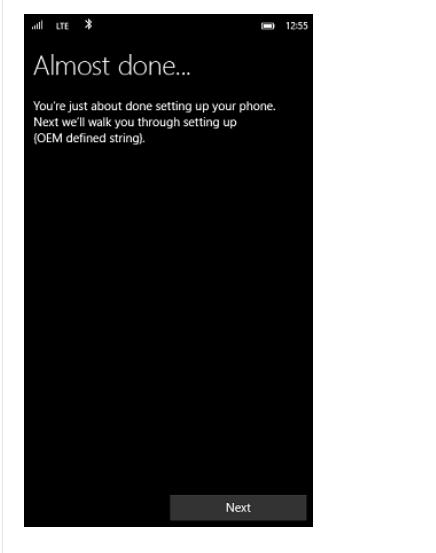
Screen 4

Screen 5

Screen 6



Screen 7



## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Screen background color during initial setup

10/2/2018 • 2 minutes to read • [Edit Online](#)

For Windows 10 Mobile, the default background during OOBE or initial device setup is always dark. To align with this change, OEMs can no longer change the default screen background color during OOBE or initial device setup.

# Set the default country/region when SIM PIN is on

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can customize the default home country/region that shows up during OOBE in cases where the SIM PIN is turned on. This value is associated with the default ICCID values. When SIM PIN is turned off, the OS uses the MCC-derived country/region instead.

If enabling this customization, OEMs can specify one or more ICCID digit prefix strings and the desired country/region to associate with the ICCID prefix. OEMs can also specify an alternative IccidToRegion.xml mapping table to use as the lookup table during device setup. This table is a tree of ICCID digit segments.

**Constraints:** ImageTimeOnly

**Instructions:**

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="SetBrandingSlot"
    Description="Use to modify the default ICCID settings."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Multivariant">
            <Setting Name="IccidToRegionOverride/$(PREFIX)" Value="" />
            <!-- Use to specify more than one ICCID digit prefix strings and their values
            <Setting Name="IccidToRegionOverride/$(PREFIX)" Value="" />
            <Setting Name="IccidToRegionOverride/$(PREFIX)" Value="" />
            <Setting Name="IccidToRegionOverride/$(PREFIX)" Value="" />
            -->
            <Setting Name="IccidToRegionTablePath" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. For `IccidToRegionOverride`:

- Replace `$(PREFIX)` with a string that you want to use for the ICCID digit prefix.
- Set the value to the desired region you want to associate with the ICCID digit prefix you added. The value must be in ISO-3166-1 Alpha-2 format.

You must do this for each ICCID prefix that you want to configure. For example, if you are adding more than one ICCID digit prefix, you must specify a different `IccidToRegionOverride` `$(PREFIX)` and the value that corresponds to that prefix.

4. To specify a different IccidToRegion.xml mapping table, set `IccidToRegionTablePath` to the path that contains the new mapping table.

The table format is a tree of ICCID digit segments and must contain the following elements:

- **IccidToRegion** - Parent element.
- **Segment** - First level of one or more segments.
- **Region** - Optional element. If the OS doesn't find a better match, it falls back to this value.
- **Digits** - Use to specify one or more digit strings that must match to continue the descent through the table.
- **Segment** - Use to specify one or more child **Segments**. ICCID matching continues after the digits that matched at the current level.

The following example shows the first few segments and digit strings in the default OS IccidToRegion.xml mapping table:

```
<?xml version="1.0" encoding="utf-8" ?>
<IccidToRegion>
  <Segment> <!-- Row 1 -->
    <!-- Fall back to US if no match -->
    <Region>US</Region>
    <Digits>01</Digits>
    <Digits>1</Digits>
    <!-- Known Issuer Identifiers -->
    <Segment><Region>AI</Region><Digits>010</Digits></Segment>
    <Segment>
      <Region>AG</Region>
      <Digits>011</Digits>
      <Digits>130</Digits>
    </Segment>
    <Segment><Region>BB</Region><Digits>012</Digits></Segment>
    <Segment>
      <Region>BM</Region>
      <Digits>013</Digits>
      <Digits>232</Digits>
      <Digits>351</Digits>
    </Segment>
    <Segment><Region>BS</Region><Digits>282</Digits></Segment>
    <Segment>
      <Region>CA</Region>
      <Digits>007</Digits>
      <Digits>035</Digits>
      <Digits>037</Digits>
      <Digits>223</Digits>
      <Digits>228</Digits>
      <Digits>236</Digits>
      <Digits>248</Digits>
      <Digits>250</Digits>
      <Digits>258</Digits>
      <Digits>277</Digits>
      <Digits>369</Digits>
      <Digits>370</Digits>
      <Digits>456</Digits>
      <Digits>482</Digits>
      <Digits>490</Digits>
      <Digits>593</Digits>
      <Digits>628</Digits>
      <Digits>629</Digits>
      <Digits>635</Digits>
      <Digits>654</Digits>
      <Digits>660</Digits>
      <Digits>682</Digits>
      <Digits>687</Digits>
      <Digits>688</Digits>
      <Digits>699</Digits>
      <Digits>727</Digits>
      <Digits>789</Digits>
      <Digits>821</Digits>
```

```

<Digits>831</Digits>
<Digits>835</Digits>
<Digits>837</Digits>
<Digits>869</Digits>
<Digits>895</Digits>
<Digits>930</Digits>
</Segment>
<Segment><Region>DM</Region><Digits>016</Digits></Segment>
<Segment>
    <Region>DO</Region>
    <Digits>028</Digits>
    <Digits>650</Digits>
</Segment>
<Segment><Region>GD</Region><Digits>017</Digits></Segment>
<Segment><Region>GU</Region><Digits>008</Digits></Segment>
<Segment>
    <Region>JM</Region>
    <Digits>018</Digits>
    <Digits>050</Digits>
    <Digits>582</Digits>
</Segment>
<Segment><Region>KN</Region><Digits>020</Digits></Segment>
<Segment><Region>KY</Region><Digits>015</Digits></Segment>
<Segment><Region>LC</Region><Digits>021</Digits></Segment>
<Segment><Region>MP</Region><Digits>025</Digits></Segment>
<Segment><Region>MS</Region><Digits>019</Digits></Segment>
<Segment>
    <Region>PR</Region>
    <Digits>283</Digits>
    <Digits>809</Digits>
    <Digits>853</Digits>
</Segment>
<Segment><Region>TC</Region><Digits>024</Digits></Segment>
<Segment><Region>TT</Region><Digits>023</Digits></Segment>
<Segment><Region>VC</Region><Digits>022</Digits></Segment>
<Segment>
    <Region>VG</Region>
    <Digits>014</Digits>
    <Digits>348</Digits>
</Segment>
</IccidToRegion>

```

## Testing steps:

1. Flash a build containing this customization to a phone. Your phone must be using a SIM that has an ICCID that matches one or more of the prefixes that you specified in the customization.
2. Go through initial device setup.
3. If you added one or more ICCID digit prefixes, verify that the home country/region has changed based on the ICCID digit prefix you had set up.

Or, if you used a different IccidToRegion.xml mapping table, verify that the home country/region that shows during initial device setup matches the value you specified in the mapping table.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Customizations for browser

10/2/2018 • 2 minutes to read • [Edit Online](#)

Describes the customizations for the browser.

## In this section

TOPIC	DESCRIPTION
<a href="#">Custom HTTP headers for Microsoft Edge</a>	Partners can configure Microsoft Edge to send custom HTTP headers, in addition to the default HTTP headers, with all HTTP and HTTPS requests. The header is the portion of the HTTP request that defines the form of the message.
<a href="#">Custom user agent string for Microsoft Edge</a>	The user agent string indicates which browser you are using, its version number, and details about your system, such as operating system and version. A web server can use this information to provide content that is tailored for your specific browser and phone.
<a href="#">Default value for browser data saver</a>	Partners can use this customization to configure the default setting for the browser data saver feature by turning the browser optimization service on or off.
<a href="#">Show pictures automatically when browsing</a>	Partners can enable or disable the Show pictures automatically setting in the browser's advanced settings screen.
<a href="#">Welcome home page for Microsoft Edge</a>	Partners can set the home page that appears the first time that Microsoft Edge is opened. This page is only shown the first time the browser is opened. After that, the browser displays either the most recently viewed page or an empty page if the user has closed all tabs or opens a new tab.
<a href="#">WinInet ReceiveTimeOut duration</a>	In cases where there are issues related to network handovers, partners can increase the WinInet ReceiveTimeOut value to provide more time for the network switch to take place.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Custom HTTP headers for Microsoft Edge

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can configure Microsoft Edge to send custom HTTP headers, in addition to the default HTTP headers, with all HTTP and HTTPS requests. The header is the portion of the HTTP request that defines the form of the message.

## **Limitations and restrictions:**

- A maximum of 16 custom headers can be defined.
- Custom headers cannot be used to modify the user agent string.
- Each header must be no more than 1 KB in length.

The following header names are reserved and must not be overwritten:

- Accept
- Accept-Charset
- Accept-Encoding
- Authorization
- Expect
- Host
- If-Match
- If-Modified-Since
- If-None-Match
- If-Range
- If-Unmodified-Since
- Max-Forwards
- Proxy-Authorization
- Range
- Referer
- TE
- USER-AGENT
- X-WAP-PROFILE

**Constraints:** FirstVariationOnly

## **Instructions:**

1. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="CustomHTTPHeaders"
    Description="Use to configure Microsoft Edge to send custom HTTP headers."
    Owner=""
    OwnerType="OEM">

    <Static>
        <Settings Path="InternetExplorer">

            <!-- Use to configure Microsoft Edge custom HTTP header 1 -->
            <Setting Name="CustomHTTPHeaders1/${ValueName}" Value="" />

            <!-- Use to configure Microsoft Edge custom HTTP header 2 -->
            <Setting Name="CustomHTTPHeaders2/${ValueName}" Value="" />

            <!-- Use to configure up to 16 Microsoft Edge custom HTTP headers
            <Setting Name="CustomHTTPHeaders3/${ValueName}" Value="" />
            <Setting Name="CustomHTTPHeaders4/${ValueName}" Value="" />
            <Setting Name="CustomHTTPHeaders5/${ValueName}" Value="" />
            <Setting Name="CustomHTTPHeaders6/${ValueName}" Value="" />
            <Setting Name="CustomHTTPHeaders7/${ValueName}" Value="" />
            <Setting Name="CustomHTTPHeaders8/${ValueName}" Value="" />
            <Setting Name="CustomHTTPHeaders9/${ValueName}" Value="" />
            <Setting Name="CustomHTTPHeaders10/${ValueName}" Value="" />
            <Setting Name="CustomHTTPHeaders11/${ValueName}" Value="" />
            <Setting Name="CustomHTTPHeaders12/${ValueName}" Value="" />
            <Setting Name="CustomHTTPHeaders13/${ValueName}" Value="" />
            <Setting Name="CustomHTTPHeaders14/${ValueName}" Value="" />
            <Setting Name="CustomHTTPHeaders15/${ValueName}" Value="" />
            <Setting Name="CustomHTTPHeaders16/${ValueName}" Value="" />
            -->

        </Settings>
    </Static>

</ImageCustomizations>

```

2. Specify an `Owner`.
3. Specify the value name for `CustomHTTPHeaders1` by replacing `$(ValueName)` with the *Header name* and set the `Value` to a *String*.

*Header name* is the unique header name and *String* is the string that the header should pass for all HTTP and HTTPS requests.

4. Specify additional custom headers, if needed. You can specify up to 16 custom HTTP headers.

#### Testing steps:

1. Flash the build containing this customization to a phone.
2. Tap on the Microsoft Edge tile to open the browser.
3. Navigate to a site that mirrors the header information for HTTP requests, and verify that your headers appear as defined.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Custom user agent string for Microsoft Edge

10/2/2018 • 2 minutes to read • [Edit Online](#)

The user agent string indicates which browser you are using, its version number, and details about your system, such as operating system and version. A web server can use this information to provide content that is tailored for your specific browser and phone.

The user agent string for the browser cannot be modified. By default, the string has the following format:

```
Mozilla/5.0 (Windows Phone 10.0; Android 4.2.1; <Manufacturer>; <Device>) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/42.0.2311.135 Mobile Safari/537.36 Edge/12.10166
```

- <Manufacturer> is automatically replaced with the OEM name. This is the same as the `PhoneManufacturer` setting value that is set as part of the customization [Phone metadata in DeviceTargetingInfo](#).
- <Device> is replaced with the device name or phone name. This is the same as the `PhonemodelName` setting value that is set as part of the customization [Phone metadata in DeviceTargetingInfo](#).

## Limitations and restrictions:

- The user agent string for the browser cannot be modified outside of the customizations listed above.
- The user agent type registry setting cannot be modified or used to change the default browser view from Mobile to Desktop.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Default value for browser data saver

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can configure the default setting for the browser data saver feature by turning the browser optimization service on or off, using the `BrowserDataSaver` setting.

**Constraints:** None

## Instructions

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="BrowserDataSaver"
    Description="Use to configure the default setting for the browser data saver
feature."
    Owner=""
    OwnerType="OEM">
    <Static>
        <Settings Path="InternetExplorer/DataSaving">
            <Setting Name="BrowserDataSaver" Value="" />
        </Settings>
    </Static>
</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set `BrowserDataSaver` to one of the following values:

VALUE	DESCRIPTION
0 or 'Disabled'	The browser data saver feature is turned off. The <b>Data Sense savings</b> option in the browser settings CPL is set to <b>off</b> .
1 or 'Enabled'	The browser data saver feature is turned on. The <b>Data Sense savings</b> option in the browser settings CPL is set to <b>automatic</b> .
Setting does not exist	The browser data saver feature is turned on. The <b>Data Sense savings</b> option in the browser settings CPL is set to <b>automatic</b> .

## Testing steps

1. Flash the build containing this customization to a device.
2. Open Microsoft Edge to launch the browser for the first time. Select **recommended** when the dialog to use the browser settings is displayed.

3. Go to the browser settings CPL.
4. Depending on the value that you set for `BrowserDataSaver`, verify:
  - If `BrowserDataSaver` is set to 0, verify that the **Data Sense savings** option is set to **off**.
  - If `BrowserDataSaver` is set to 1, verify that the **Data Sense savings** option is set to **automatic**.
  - If `BrowserDataSaver` setting has not been set, verify that the **Data Sense savings** option is set to **automatic**.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Show pictures automatically when browsing

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can enable or disable the **Show pictures automatically** setting in the browser's **advanced settings** screen.

**Constraints:** None

## Instructions

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="ShowPicturesAutomatically"
    Description="Use to enable or disable the 'Show pictures automatically' setting in
the browser's advanced settings screen."
    Owner=""
    OwnerType="OEM">
    <Static>
        <Settings Path="MicrosoftEdge/DataSaving">
            <Setting Name="ShowPicturesAutomatically" Value="" />
        </Settings>
    </Static>
</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set `ShowPicturesAutomatically` to one of the following values:

VALUE	DESCRIPTION
1 or 'Enabled'	Shows the <b>Show pictures automatically</b> option in the browser <b>advanced settings</b> screen.
0 or 'Disabled'	Disables the <b>Show pictures automatically</b> option in the browser <b>advanced settings</b> screen.

## Testing steps

1. Flash the build containing this customization to a device.
2. Open the browser settings screen and choose the **advanced settings** option.
3. From the advanced settings screen, verify that **Show pictures automatically** is either enabled or disabled depending on the value that you set for `ShowPicturesAutomatically`.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Welcome home page for Microsoft Edge

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can set the home page that appears the first time that Microsoft Edge is opened. This page is only shown the first time the browser is opened. After that, the browser displays either the most recently viewed page or an empty page if the user has closed all tabs or opens a new tab.

**Constraints:** FirstVariationOnly

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample or use the sample WelcomeHomePage.xml file.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="WelcomeHomePage"
    Description="Use to set the home page that appears the first time that Microsoft
    Edge is opened."
    Owner=""
    OwnerType="OEM">

    <Static>
        <Settings Path="InternetExplorer">
            <Setting Name="FirstRunUrl" Value="" />
        </Settings>
    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.
3. Specify the `FirstRunUrl``Value` with a valid link that starts with `http://`. It is recommended that you use a forward link that redirects the user to a localized page.

## Testing steps:

1. Flash a build containing this customization to a phone with a data connection or Wi-Fi connection enabled.
2. Open Microsoft Edge, and verify that the correct page appears.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# WinInet ReceiveTimeOut duration

10/2/2018 • 2 minutes to read • [Edit Online](#)

In cases where there are issues related to network handovers, partners can increase the WinInet ReceiveTimeOut value to provide more time for the network switch to take place.

**Constraints:** ImageTimeOnly

**Instructions:**

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="WinInetReceiveTimeOut"
    Description="Use to configure the WinInet Internet options ReceiveTimeOut value."
    Owner=""
    OwnerType="OEM">

    <Static>
        <Settings Path="WinInet/InternetSettings">
            <Setting Name="WinInetReceiveTimeOut" Value="" />
        </Settings>
    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.
3. Set the `Value`, in milliseconds, to a number between 30000 and 120000 (inclusive). This value will be used as the WinInet ReceiveTimeOut value.

The default OS value is 60000 milliseconds (60 seconds).

**Testing steps:**

Work with your mobile operator partner to fully test this customization on their network.

1. Flash a build containing this customization to a phone with a cellular connection.
2. Place the phone in an area with a weak field signal, for example a weak 3G field.
3. Download a file during 2G to 3G handover.
4. Download a file during 3G to 2G handover.
5. Verify that the files were downloaded successfully. If not, adjust the value for `WinInetReceiveTimeOut` as needed.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Customizations for cellular connectivity

10/2/2018 • 2 minutes to read • [Edit Online](#)

Describes the customizations for configuring connectivity settings.

## In this section

TOPIC	DESCRIPTION
<a href="#">Background cellular data restriction</a>	To meet market or mobile operator requirements, OEMs can restrict background data in the data usage settings.
<a href="#">Cellular data connection icon</a>	The one-, two-, or three-character codes used to signify the data connection type in the status bar can be modified.
<a href="#">Custom percentages for signal strength bars</a>	Partners must modify the percentage values used for the signal strength bars in the status bar.
<a href="#">Data transfer indicator</a>	OEMs can display a data transfer indicator on a device's status bar for mobile operators that require it.
<a href="#">Default highest connection speed</a>	Partners can set the default value for the Highest connection speed option in the Settings > Cellular & SIM > SIM screen by specifying the bitmask for any combination of radio technology to be excluded from the default value. The connection speed that has not been excluded will show up as the highest connection speed.
<a href="#">Default roaming option</a>	Partners can set the default value for the Default roaming options option in the Cellular & SIM settings screen.
<a href="#">Disable Cell Broadcast</a>	By default, Cell Broadcast (also known as Short Message Service-Cell Broadcast (SMS-CB)) is a feature that is active at all times.
<a href="#">Extended error messages for reject codes</a>	When a reject code is sent by the network, partners can specify that extended error messages should be displayed instead of the standard simple error messages. This customization is intended for use only when required by the mobile operator's network.
<a href="#">Hide CDMA mode selection</a>	For CDMA phones, partners can hide CDMA option in the network Mode selection drop-down that appears on the Cellular & SIM screen in Settings.
<a href="#">Hide Cellular &amp; SIM Settings</a>	OEMs can hide certain user options for phones that appear in the Cellular & SIM screen in Settings. These options include: Network Mode selection drop-down for World mode, Network Selection drop-down for GSM, and Network Type drop-down for CDMA.

Topic	Description
<a href="#">LTE attach: GUID for user configured internet APN</a>	Partners can set the OEMConnectionId that is used when creating the user-configured connection for internet from the SIM settings screen.
<a href="#">LTE attach: Mapping OEMConnectionId values to modem profiles</a>	Partners can set the list of OEMConnectionId values that map to an LTE attach profile in the mobile broadband driver.
<a href="#">Manual network selection timeout</a>	OEMs can change the default network selection timeout value. By default, the OS allows the device to register on the manually selected network for 60 seconds (or 1 minute) before it switches back to automatic mode.
<a href="#">Maximum number of PDP contexts</a>	OEMs can set different maximum values for the number of PDP contexts for the device if required by their mobile operator.
<a href="#">Permanent automatic mode</a>	OEMs can enable permanent automatic mode for mobile networks that require the cellular settings to revert to automatic network selection after the user has manually selected another network when roaming or out of range of the home network.
<a href="#">Preferred data provider list</a>	For mobile operators that require it, OEMs can set a list of MCC/MNC pairs for the purchase order (PO) carrier or primary operator so that it can be set as the default data line for phones that have a dual SIM.
<a href="#">Roaming filter</a>	Partners can add roaming filters that determine when the device appears to be roaming, based on the network the device is currently connected to. With roaming filters enabled, connections on other companies' specified networks are not treated as roaming.

## Related topics

[Customizations for Wi-Fi settings and Connectivity](#)

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Background cellular data restriction

10/2/2018 • 2 minutes to read • [Edit Online](#)

To meet market or mobile operator requirements, OEMs can restrict background data in the data usage settings.

OEMs can set the default value to either never restrict usage of the data plan or restrict background data when roaming.

**Constraints:** ImageTimeOnly

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>

<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="DataSaverMode"
    Description="Use to restrict background data. OEMs can set the default value to
either never restrict usage
of the data plan or restrict background data when roaming."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="DataSense">
            <Setting Name="DataSaverMode" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set the value of `DataSaverMode` to one of the following:

VALUE	DESCRIPTION
0 or 'NeverRestrict'	Never restrict usage of the data plan.
2 or 'RestrictWhenRoaming'	Restrict background data when roaming.

## Testing steps:

1. Flash the build that contains this customization to a device.
2. Go to the **Data usage** settings screen.

Verify that the correct settings option is enabled depending on the default value that you set. A toggle to restrict background data also becomes available to the user.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Cellular data connection icon

10/2/2018 • 2 minutes to read • [Edit Online](#)

The one-, two-, or three-character codes used to signify the data connection type in the status bar can be modified. The default values are G (GPRS), 1x (RTT), DO (EVDO), E (EDGE), 3G (3G), H (HSDPA/HSUPA), LTE (LTE), or no letters displayed if there is no active connection.

## Limitations and restrictions:

- Partners cannot modify the types of data connections available; only the display code can be modified.

## Constraints:

- Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="DataConnectionIcon"
    Description="Use to modify the one-, two-, or three-character codes used to signify
the data connection type in the status bar."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Shell/SystemTray/DataConnectionString">
            <Setting Name="1XRTTDEFAULT" Value="1X" />
            <Setting Name="1XRTT" Value="1X" />
            <Setting Name="EVDODEFAULT" Value="DO" />
            <Setting Name="EVDOREV0" Value="DO" />
            <Setting Name="EVDOREVA" Value="DO" />
            <Setting Name="EVDOREV0" Value="DO" />
            <Setting Name="GSMDEFAULT" Value="G" />
            <Setting Name="GSMGSM" Value="" />
            <Setting Name="GSMGPRS" Value="G" />
            <Setting Name="GSMEDGE" Value="E" />
            <Setting Name="UMTSDEFAULT" Value="3G" />
            <Setting Name="UMTSUMTS" Value="3G" />
            <Setting Name="UMTSHSDPA" Value="H" />
            <Setting Name="UMTSHSUPA" Value="H" />
            <Setting Name="UMTSHPAPLUS" Value="H+" />
            <Setting Name="UMTSCHSPAPLUS" Value="H+" />
            <Setting Name="UMTSHPAPLUS64QAM" Value="H+" />
            <Setting Name="LTEDEFAULT" Value="LTE" />
            <Setting Name="LTFDD" Value="LTE" />
            <Setting Name="LTETDD" Value="LTE" />
            <Setting Name="TDSCDMADEFAULT" Value="T" />
            <Setting Name="TDSCDMAUMTS" Value="T" />
            <Setting Name="TDSCDMAHSDPA" Value="H" />
            <Setting Name="TDSCDMAHSUPA" Value="H" />
            <Setting Name="TDSCDMAHPAPLUS" Value="H+" />
            <Setting Name="TDSCDMADCHSPAPLUS" Value="H+" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. To change the default one-, two-, or three-character codes used to signify the data connection type in the status bar, modify the `value` that corresponds to the setting you want to change.

**NOTE**

All the `value` attributes are the current default values for each data connection type. You can change these to something else. Maximum character length is 3 characters.

The following table shows the settings you can modify and their default values.

SETTING NAME	DEFAULT VALUE	DESCRIPTION
1XRTTDEFAULT	1X	1XRTT connection
1XRTT	1X	1XRTT
EVDODEFAULT	DO	EVDO connection
EVDOREV0	DO	EVDO rev. 0
EVDOREVA	DO	EVDO rev. A
EVDOREVB	DO	EVDO rev. B
GSMDEFAULT	G	GSM connection
GSMGSM		No GSM connection
GSMGPRS	G	GSM General Packet Radio Service
GSMEDGE	E	GSM Enhanced Data rates for Global Evolution
UMTSDEFAULT	3G	UMTS connection
UMTSLTE	3G	UMTS
UMTSHSDPA	H	High Speed Downlink Packet Access
UMTSHSUPA	H	High Speed Uplink Packet Access
UMTSHSPAPLUS	H+	High Speed Packet Access "Plus"
UMTSDCHSPAPLUS	H+	Dual-carrier HSPA+
UMTSHSPAPLUS64QAM	By default, the value inherited from UMTSDCHSPAPLUS.  To set a value different from the value for UMTSDCHSPAPLUS, you must set the value explicitly.	UMTS HSPA+ 64QAM (high order modulation)

SETTING NAME	DEFAULT VALUE	DESCRIPTION
LTEDEFAULT	LTE	LTE connection
LTEFDD	LTE	LTE Frequency Division Duplexing
LTETDD	LTE	LTE Time Division Duplexing
TDSCDMADEFAULT	T	TD-SCDMA connection
TDSCDMAUMTS	T	TD-SCDMA
TDSCDMAHSDPA	H	TD-SCDMA High Speed Downlink Packet Access
TDSCDMAHSUPA	H	TD-SCDMA High Speed Uplink Packet Access
TDSCDMAHSPAPLUS	H+	TD-SCDMA High Speed Packet Access "Plus"
TDSCDMADCHSPAPLUS	H+	TD-SCDMA Dual-carrier HSPA+

### Testing:

1. Flash an image containing this customization to a phone that has a UICC.
2. Verify that the one-, two-, or three-character code(s) you used for the cellular data connection type is displayed in the status bar at the top of the screen.

You may have to tap the clock to make the status bar appear if it is hidden.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Connection speed option

10/2/2018 • 5 minutes to read • [Edit Online](#)

Partners can customize the listed names of the connection speeds, and can hide the user option to select the connection speed that is displayed on the **SIM** screen.

Partners can hide the user option to select the connection speed that is displayed on the **SIM** screen in **Settings > Cellular & SIM**, if they do not want users to be able to deselect the highest possible speed.

Alternately, partners can customize the listed names of the connection speeds with their own character codes. The default values are 2G, 3G, and 4G.

**Constraints:** None

This customization supports: **per-IMSI** value, **per-device** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="ConnectionSpeedOption"
    Description="Use to hide the connection speed option or customize the default
character codes for connection speed."
    Owner=""
    OwnerType="OEM">
```

```
<!-- Use for the per-IMSI case

<!-- Define the Targets -->
<Targets>
    <Target Id="">
        <TargetState>
            <Condition Name="" Value="" />
            <Condition Name="" Value="" />
        </TargetState>
    </Target>
</Targets>

<Static>
    <Settings Path="Multivariant">
        <Setting Name="Enable" Value="1" />
    </Settings>
    <Settings Path="AutoDataConfig">
        <Setting Name="Enable" Value="0" />
    </Settings>
</Static>

<!-- Specify the Variant -->
<Variant Name="">
    <TargetRefs>
        <TargetRef Id="" />
    </TargetRefs>

    <Settings Path="CellCore/PerIMSI/$(__IMSI)/CellUX">

        <!-- Settings to hide some options or the entire Highest connection speed dropdown -->
        <!-- To hide the 4G Only option from the Highest connection speed dropdown -->
```

```


<Setting Name="HideHighestSpeed4GOnly" Value="1" />
-->

<Setting Name="HideHighestSpeed4G" Value="1" />
-->

<Setting Name="HideHighestSpeed3GOnly" Value="1" />
-->

<Setting Name="HideHighestSpeed2G" Value="1" />
-->


<Setting Name="HideHighestSpeed" Value="1" />
-->


<Setting Name="ShowHideHighestSpeed3GPreferred" Value="1" />
-->

<!-- Settings to customize the default character codes for connection speed. For example, to change the
default '4G' character
      code for highestSpeed4G, change the value to another character code such as 'LTE'.
&lt;Setting Name="HighestSpeed2G" Value="" /&gt;
&lt;Setting Name="HighestSpeed3G" Value="" /&gt;
&lt;Setting Name="HighestSpeed3GOnly" Value="" /&gt;
&lt;Setting Name="HighestSpeed3GPreferred" Value="" /&gt;
&lt;Setting Name="HighestSpeed4G" Value="" /&gt;
&lt;Setting Name="HighestSpeed4GOnly" Value="" /&gt;
--&gt;

<!-- To modify the Highest connection speed dropdown title
&lt;Setting Name="HighestSpeedTitle" Value="" /&gt;
--&gt;

&lt;/Settings&gt;
&lt;/Variant&gt;

--&gt;

<!-- Use for the per-device case

&lt;Static&gt;
  &lt;Settings Path="CellCore/PerDevice/CellUX"&gt;

    <!-- Settings to hide some options or the entire Highest connection speed dropdown --&gt;
    <!-- To hide the 4G Only option from the Highest connection speed dropdown
    &lt;Setting Name="HideHighestSpeed4GOnly" Value="1" /&gt;
    --&gt;
    <!-- To hide the 4G option from the Highest connection speed dropdown
    &lt;Setting Name="HideHighestSpeed4G" Value="1" /&gt;
    --&gt;
    <!-- To hide the 3G Only option from the Highest connection speed dropdown
    &lt;Setting Name="HideHighestSpeed3GOnly" Value="1" /&gt;
    --&gt;
    <!-- To hide the 2G option from the Highest connection speed dropdown
    &lt;Setting Name="HideHighestSpeed2G" Value="1" /&gt;
    --&gt;

    <!-- To hide the Highest connection speed dropdown entirely
    &lt;Setting Name="HideHighestSpeed" Value="1" /&gt;
    --&gt;

    <!-- To show the 3G Preferred option in the Highest connection speed dropdown
    &lt;Setting Name="ShowHideHighestSpeed3GPreferred" Value="1" /&gt;
    --&gt;

    <!-- Settings to customize the default character codes for connection speed. For example, to change the
    default '4G' character
</pre>

```

```

default 4G character
    code for highestSpeed4G, change the value to another character code such as 'LTE'.
<Setting Name="HighestSpeed2G" Value="" />
<Setting Name="HighestSpeed3G" Value="" />
<Setting Name="HighestSpeed3GOnly" Value="" />
<Setting Name="HighestSpeed3GPreferred" Value="" />
<Setting Name="HighestSpeed4G" Value="" />
<Setting Name="HighestSpeed4GOnly" Value="" />
-->

<!-- To modify the Highest connection speed dropdown title
<Setting Name="HighestSpeedTitle" Value="" />
-->

</Settings>
</Static>

-->

</ImageCustomizations>
```

```

1. Specify an `Owner`.
2. Determine if you need to use the **per-IMSI** or **per-device** setting.

For the **per-IMSI** case:

- a. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
- b. Define settings for a **Variant**, which are applied if the associated target's conditions are met.

### 3. To hide some options or the entire Highest connection speed drop-down

- To hide the "4G Only" option from the **Highest connection speed** drop-down in the **Settings > Cellular & SIM > SIM** screen, set `HideHighestSpeed4GOnly` to 1 or 'Yes'.
- To hide the "4G" option from the **Highest connection speed** drop-down in the **Settings > Cellular & SIM > SIM** screen, set `HideHighestSpeed4G` to 1 or 'Yes'.
- To hide the "3G Only" option from the **Highest connection speed** drop-down in the **Settings > Cellular & SIM > SIM** screen, set `HideHighestSpeed3GOnly` to 1 or 'Yes'.
- To hide the "2G" option from the **Highest connection speed** drop-down in the **Settings > Cellular & SIM > SIM** screen, set `HideHighestSpeed2G` to 1 or 'Yes'.
- To hide the **Highest connection speed** drop-down entirely, set `HideHighestSpeed` to 1 or 'Yes'.

### 4. To show the 3G Preferred option in the Highest connection speed drop-down

- Set `ShowHighestSpeed3GPreferred` to 1 or 'Yes'.

### 5. To customize the character codes for connection speed

- To modify the "2G" string to another character code, change the `HighestSpeed2G``Value`.
- To modify the "3G" string to another character code, change the `HighestSpeed3G``Value`.
- To modify the "3G Only" string to another character code, change the `HighestSpeed3GOnly``Value`.
- To modify the "4G (3G Preferred)" string to another character code, change the `HighestSpeed3GPreferred``Value`.
- To modify the "4G" string to another character code, change the `HighestSpeed4G``Value`.

- To modify the "4G Only" string to another character code, change the `HighestSpeed4GOnly``Value`.

For example, to change the default '4G' character code for `HighestSpeed4G` to another character code such as 'LTE', set the `Value` to 'LTE'. Although there is no limit to number of characters you can use, if the character code is too long, this will be truncated in the UI.

#### Note

You must include all three values (even if you are only modifying one) or the display text will not be set correctly.

## 6. To customize the Highest connection speed drop-down label

- Set the `HighestSpeedTitle``Value` to the string that you want to use.

For example, to change 'Highest connection speed' to another string such as 'Preferred connection speed', set the `Value` to 'Preferred connection speed'.

#### Testing:

1. Flash the build containing this customization to a device.
2. Go to the **Settings > Cellular & SIM > SIM** screen.
3. If you hid the **Highest connection speed** drop-down or one or more options within the drop-down, verify that the behavior matches your setting.
4. If you customized the dropdown label, or the one- or two-character codes for connection speed for one or more options within the connection speed drop-down, verify that the string matches what you specified.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Custom percentages for signal strength bars

10/2/2018 • 3 minutes to read • [Edit Online](#)

Partners must modify the percentage values used for the signal strength bars in the status bar. These five bars represent the strength of the cellular connection and are determined by a mapping table defined for the network. It is considered a manufacturing defect if a device ships with incomplete or incorrect mapping tables.

Partners must also tune the thresholds for measuring the signal strength received from the modem. Filters determine the magnitude of change in signal strength that triggers a report. Filter values are specific to the mobile operator network. The following filters must be configured:

- GSM
- UMTS RSSI
- UMTS Ec/No
- CDMA 1X Receive Channel Power
- CDMA 1X Pilot Energy
- CDMA EVDO Carrier Strength
- CDMA EVDO SINR
- LTE reference signal received power (RSRP)
- LTE reference signal signal-to-noise ratio (RS\_SNR)

## Note

Windows 10 Mobile always shows the signal strength for the highest technology. For example:

- If 1X and EVDO register at the same time, the device shows the EVDO signal strength in the status bar.
- If 1X and LTE register at the same time, the device shows the LTE signal strength in the status bar.

In these examples, if the 1X signal strength changes and there is a higher radio access technology, the device will not do anything.

## Constraints:

This customization supports: **per-device** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="SignalStrengthBars"
    Description="Use to modify the percentage values used for the signal strength bars
    in the status bar."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Cellcore/PerDevice/External">
            <!-- Use to modify the percentage value for GSM. The numbers represent the signal strength bars. -->
        </Settings>
    </Static>
</ImageCustomizations>
```

```

<Setting Name="SignalBarMappingTable/GERAN/5" Value="" />
<Setting Name="SignalBarMappingTable/GERAN/4" Value="" />
<Setting Name="SignalBarMappingTable/GERAN/3" Value="" />
<Setting Name="SignalBarMappingTable/GERAN/2" Value="" />
<Setting Name="SignalBarMappingTable/GERAN/1" Value="" />

<!-- Use to modify the percentage value for UMTS RSSI. The numbers represent the signal strength bars. -->
<Setting Name="SignalBarMappingTable/UMTS/5" Value="" />
<Setting Name="SignalBarMappingTable/UMTS/4" Value="" />
<Setting Name="SignalBarMappingTable/UMTS/3" Value="" />
<Setting Name="SignalBarMappingTable/UMTS/2" Value="" />
<Setting Name="SignalBarMappingTable/UMTS/1" Value="" />

<!-- Use to modify the percentage value for UMTS Ec/No. The numbers represent the signal strength bars. -->
<Setting Name="SignalBarMappingTable/UMTSEcNo/5" Value="" />
<Setting Name="SignalBarMappingTable/UMTSEcNo/4" Value="" />
<Setting Name="SignalBarMappingTable/UMTSEcNo/3" Value="" />
<Setting Name="SignalBarMappingTable/UMTSEcNo/2" Value="" />
<Setting Name="SignalBarMappingTable/UMTSEcNo/1" Value="" />

<!-- Use to modify the percentage value for CDMA 1X Receive Channel Power. The numbers represent the signal strength bars. -->
<Setting Name="SignalBarMappingTable/CDMA1xRCP/5" Value="" />
<Setting Name="SignalBarMappingTable/CDMA1xRCP/4" Value="" />
<Setting Name="SignalBarMappingTable/CDMA1xRCP/3" Value="" />
<Setting Name="SignalBarMappingTable/CDMA1xRCP/2" Value="" />
<Setting Name="SignalBarMappingTable/CDMA1xRCP/1" Value="" />

<!-- Use to modify the percentage value for CDMA 1X Pilot Energy. The numbers represent the signal strength bars. -->
<Setting Name="SignalBarMappingTable/CDMA1xPE/5" Value="" />
<Setting Name="SignalBarMappingTable/CDMA1xPE/4" Value="" />
<Setting Name="SignalBarMappingTable/CDMA1xPE/3" Value="" />
<Setting Name="SignalBarMappingTable/CDMA1xPE/2" Value="" />
<Setting Name="SignalBarMappingTable/CDMA1xPE/1" Value="" />

<!-- Use to modify the percentage value for CDMA EVDO Carrier Strength. The numbers represent the signal strength bars. -->
<Setting Name="SignalBarMappingTable/CDMAEvdoCS/5" Value="" />
<Setting Name="SignalBarMappingTable/CDMAEvdoCS/4" Value="" />
<Setting Name="SignalBarMappingTable/CDMAEvdoCS/3" Value="" />
<Setting Name="SignalBarMappingTable/CDMAEvdoCS/2" Value="" />
<Setting Name="SignalBarMappingTable/CDMAEvdoCS/1" Value="" />

<!-- Use to modify the percentage value for CDMA EVDO SINR. The numbers represent the signal strength bars. -->
<Setting Name="SignalBarMappingTable/CDMAEvdoSINR/5" Value="" />
<Setting Name="SignalBarMappingTable/CDMAEvdoSINR/4" Value="" />
<Setting Name="SignalBarMappingTable/CDMAEvdoSINR/3" Value="" />
<Setting Name="SignalBarMappingTable/CDMAEvdoSINR/2" Value="" />
<Setting Name="SignalBarMappingTable/CDMAEvdoSINR/1" Value="" />

<!-- Use to modify the percentage value for LTE reference signal received power. The numbers represent the signal strength bars. -->
<Setting Name="SignalBarMappingTable/LTESRSRP/5" Value="" />
<Setting Name="SignalBarMappingTable/LTESRSRP/4" Value="" />
<Setting Name="SignalBarMappingTable/LTESRSRP/3" Value="" />
<Setting Name="SignalBarMappingTable/LTESRSRP/2" Value="" />
<Setting Name="SignalBarMappingTable/LTESRSRP/1" Value="" />

<!-- Use to modify the percentage value for LTE reference signal signal-to-noise ratio. The numbers represent the signal strength bars. -->
<Setting Name="SignalBarMappingTable/LTERSSNR/5" Value="" />
<Setting Name="SignalBarMappingTable/LTERSSNR/4" Value="" />
<Setting Name="SignalBarMappingTable/LTERSSNR/3" Value="" />
<Setting Name="SignalBarMappingTable/LTERSSNR/2" Value="" />
<Setting Name="SignalBarMappingTable/LTERSSNR/1" Value="" />

```

```
</Settings>  
</Static>  
</ImageCustomizations>
```

2. Specify an `Owner`.
3. Set the `Value` for the filters that you want to configure. You can use either a decimal or hexadecimal value, but you must add the `0x` prefix when specifying a hexadecimal value.

**Note**

The signal strength mapping is implemented by the modem vendor. For information about how to change the required values under the **SignalBarMappingTable** setting, see the documentation provided by the modem vendor.

**Testing:**

Refer to the documentation provided by the modem vendor and work with your mobile operator to test this customization on their network.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Data transfer indicator

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can display a data transfer indicator on a device's status bar for mobile operators that require it. When this feature is enabled, an arrow is displayed above the cellular data connection icon or Wi-Fi connection icon to indicate that data transfer is occurring.

The data transfer indicator, the cellular data connection icon, and the cellular signal strength icon are promoted on the status bar for 10 seconds and does not appear more frequently than once every 5 minutes. The data transfer indicator and the Wi-Fi connection icon are not promoted on the status bar.

However, users can tap the status bar to view the data transfer indicator above the bearer that's transmitting data. The data transfer indicator, the cellular data connection icon, and the cellular signal strength icon are displayed for 10 seconds if cellular data transfer is occurring. The data transfer indicator and the Wi-Fi connection icon are displayed for 2 seconds if Wi-Fi data transfer is occurring.

**Constraints:** None

**Instructions:**

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="DataTransferIndicator"
    Description="Use to display a data transfer indicator on a device's status bar for
operators that require it."
    Owner=""
    OwnerType="OEM">

    <Static>
        <Settings Path="Shell/SystemTray/DataActivity">
            <Setting Name="DataActivityIcon" Value="" />
        </Settings>
    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set `DataActivityIcon` to one of the following values:

VALUE	DESCRIPTION
0 or 'Disabled'	Hides the data transfer indicator in the status bar. This is the default behavior if the setting is not set.
1 or 'Enabled'	Shows the data transfer indicator in the status bar.

**Testing:**

1. Flash an image that contains this customization to a device.

2. Turn off Wi-Fi and send data over your cellular connection. For example, send an email with a photo attachment. Verify that the arrow that indicates data transfer appears above the cellular data connection icon on the status bar.
3. Turn on Wi-Fi and send data over your Wi-Fi connection. Verify that the arrow that indicates data transfer appears above the Wi-Fi connection icon on the status bar.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Default highest connection speed

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can set the default value for the **Highest connection speed** option in the **Settings > Cellular & SIM > SIM** screen by specifying the bitmask for any combination of radio technology to be excluded from the default value. The connection speed that has not been excluded will show up as the highest connection speed.

Users can later change the highest connection speed setting on the device.

## Note

On dual SIM devices that only support up to 3G connection speeds, the **Highest connection speed** option is replaced by a 3G on/off toggle based on the per-device setting. **On** means that 3G is preferred and **Off** means 2G only.

**Constraints:** None

This customization supports: **per-IMSI** value, **per-device** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="DefaultHighestConnectionSpeed"
    Description="Use to set the default value for the highest connection speed in the
    cellular Settings CPL."
    Owner=""
    OwnerType="OEM">

<!-- Use for the per-IMSI case

    &lt;!-- Define the Targets --&gt;
    &lt;Targets&gt;
        &lt;Target Id=""&gt;
            &lt;TargetState&gt;
                &lt;Condition Name="" Value="" /&gt;
                &lt;Condition Name="" Value="" /&gt;
            &lt;/TargetState&gt;
        &lt;/Target&gt;
    &lt;/Targets&gt;

    &lt;Static&gt;
        &lt;Settings Path="Multivariant"&gt;
            &lt;Setting Name="Enable" Value="1" /&gt;
        &lt;/Settings&gt;
        &lt;Settings Path="AutoDataConfig"&gt;
            &lt;Setting Name="Enable" Value="0" /&gt;
        &lt;/Settings&gt;
    &lt;/Static&gt;

    &lt;!-- Specify the Variant --&gt;
    &lt;Variant Name=""&gt;
        &lt;TargetRefs&gt;
            &lt;TargetRef Id="" /&gt;
        &lt;/TargetRefs&gt;

        &lt;Settings Path="CellCore/PerIMSI/$(__IMSI)/General"&gt;
            &lt;Setting Name="ExcludedSystemTypesByDefault" Value="" /&gt;
        &lt;/Settings&gt;
    &lt;/Variant&gt;

--&gt;

    &lt;!-- Use for the per-device case

    &lt;Static&gt;
        &lt;Settings Path="CellCore/PerDevice/General"&gt;
            &lt;Setting Name="ExcludedSystemTypesByDefault" Value="" /&gt;
        &lt;/Settings&gt;
    &lt;/Static&gt;

--&gt;

&lt;/ImageCustomizations&gt;
</pre>

```

2. Specify an **Owner**.

3. Determine if you need to use the **per-IMSI** or **per-device** setting.

For the **per-IMSI** case:

- Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
- Define settings for a **Variant**, which are applied if the associated target's conditions are met.

4. Specify the **ExcludedSystemTypesByDefault``Value** to set a default value for the **Highest connection speed**

option in the **Settings > Cellular** screen.

- a. Refer to [RILSYSTEMTYPE] and note the values for the corresponding radio technology that you want to exclude.

For example, on an LTE network the default setting for the highest connection speed is 4G. The other available options that show up also include 3G and 2G. However, if you want to change the default to 2G, you will need to exclude RIL\_SYSTEMTYPE\_LTE (4G) and RIL\_SYSTEMTYPE\_UMTS (3G) to set the default to 2G. To do this, note the values for RIL\_SYSTEMTYPE\_LTE (4G) and RIL\_SYSTEMTYPE\_UMTS (3G) in hexadecimal and convert these to binary.

	HEXADECIMAL	BINARY
RIL_SYSTEMTYPE_LTE (4G)	0x10	10000
RIL_SYSTEMTYPE_UMTS (3G)	0x8	01000

- b. Perform a bitwise **OR** operation on the radio technologies you want to exclude.

For example, a bitwise **OR** operation on RIL\_SYSTEMTYPE\_LTE (4G) and RIL\_SYSTEMTYPE\_UMTS (3G) results in the value 11000 (binary) or 0x18 (hexadecimal). This means that for this example, `ExcludedSystemTypesByDefault``Value` must be set to 0x18 to change the default highest connection speed to 2G.

Partners should note that there is no 3G only option for the highest connection speed. The architecture is designed such that 3G means 3G is preferred and 2G is allowed.

#### Testing:

1. Flash the build containing this customization to a device.
2. Go to the **Settings > Cellular & SIM > SIM** screen.
3. Verify that the **Highest connection speed** shows the correct default value that you set.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Default roaming option

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can set the default value for the **Default roaming options** option in the **Cellular & SIM** settings screen.

Users can later change the default roaming option on the device.

**Constraints:** None

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="DefaultRoamingOption"
    Description="Use to set the default roaming option."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Connections/General">
            <Setting Name="DataRoam" Value="" />
        </Settings>

    </Static>
</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set the value of `DataRoam` to one of the following:

VALUE	OPTION
0 or 'DoNotRoam'	Don't roam
1 or 'DomesticRoaming'	Don't roam (domestic roaming if applicable)
2 or 'InternationalRoaming'	Roam (international roaming if applicable)

## Testing:

1. Flash the build containing this customization to a device.
2. Go to the **Settings > Cellular & SIM** screen.
3. Verify that the **Default roaming options** shows the correct default value that you set.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Disable Cell Broadcast

10/2/2018 • 2 minutes to read • [Edit Online](#)

By default, Cell Broadcast (also known as Short Message Service-Cell Broadcast (SMS-CB)) is a feature that is active at all times. Some mobile operators may require OEMs to disable this feature if it is not available for certain areas or zones, or if operators want to get better battery performance by not activating radio functions that may not be needed for a certain market.

To comply with these operator requirements, OEMs may disable Cell Broadcast through an NV item setting in the modem rather than in the OS.

## Note

NV items are owned by the IHV and not Microsoft. OEMs should consult with their IHV to determine how to disable Cell Broadcast through an NV item.

# Extended error messages for reject codes

10/2/2018 • 2 minutes to read • [Edit Online](#)

When a reject code is sent by the network, partners can specify that extended error messages should be displayed instead of the standard simple error messages. This customization is intended for use only when required by the mobile operator's network.

The short versions of the extended reject message are shown in the following screens:

- **Phone** tile in **Start**
- **Call History** screen
  - Dialer
- **Call Progress** screen
- **Incoming Call** screen
- As the status string under **Settings > Cellular & SIM**

The long version of the extended reject message is shown in the following screen:

- Under the **Active Network** label in **Settings > Cellular & SIM**

The OS handles three extended reject codes:

REJECT CODE	LONG VERSION	SHORT VERSION
2 (The SIM card hasn't been activated or has been deactivated)	SIM not set up MM#2	Invalid SIM
3 (The SIM card fails authentication or one of the identity check procedures. This can also happen due to a duplication of the TMSI across different MSCs)	Can't verify SIM MM#3	Invalid SIM
6 (The device has been put on a block list, such as when the device has been stolen or the IMEI is restricted)	Phone not allowed MM#6	No Service

**Constraints:** None

This customization supports: **per-IMSI** value, **per-device** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="ExtendedRejectCodes"
    Description="Use to specify that extended error messages should be displayed
    instead of standard simple messages."
    Owner=""
    OwnerType="OEM">

```

<!-- Use for the per-IMSI case

```

<!-- Define the Targets -->
<Targets>
    <Target Id="">
        <TargetState>
            <Condition Name="" Value="" />
            <Condition Name="" Value="" />
        </TargetState>
    </Target>
</Targets>

<Static>
    <Settings Path="Multivariant">
        <Setting Name="Enable" Value="1" />
    </Settings>
    <Settings Path="AutoDataConfig">
        <Setting Name="Enable" Value="0" />
    </Settings>
</Static>

<!-- Specify the Variant -->
<Variant Name="">
    <TargetRefs>
        <TargetRef Id="" />
    </TargetRefs>

    <Settings Path="CellCore/PerIMSI/$(__IMSI)/CellUX">
        <Setting Name="ShowExtendedRejectCodes" Value="" />
    </Settings>
</Variant>

-->

<!-- Use for the per-device case

<Static>
    <Settings Path="CellCore/PerDevice/CellUX">
        <Setting Name="ShowExtendedRejectCodes" Value="" />
    </Settings>
</Static>

-->

</ImageCustomizations>
```

```

1. Specify an `Owner`.

2. Set the `Branding flags` setting in the [Branding for phone calls](#) customization so that the `ExtendedRejectCodes` flag is enabled.

#### Note

The `ExtendedRejectCodes` flag is not enabled by default so make sure that this is set. Both the

`ShowExtendedRejectCodes` setting and ExtendedRejectCodes flag need to be set for the customization to be fully enabled.

### 3. Determine if you need to use the **per-IMSI** or **per-device** setting.

For the **per-IMSI** case:

a. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.

b. Define settings for a **Variant**, which are applied if the associated target's conditions are met.

### 4. Set the value for `ShowExtendedRejectCodes` to one of the following:

| VALUE      | DESCRIPTION   |
|------------|---|
| 0 or 'No'  | Hides the extended error messages when devices receive LAU reject codes with cause number 2, 3, or 6. |
| 1 or 'Yes' | Shows the extended error messages when devices receive LAU reject codes with cause number 2, 3, or 6. |

The default for this setting is to show the \*\*CDMA\*\* option in the \*\*Mode\*\* selection drop-down that appears in the \*\*Cellular & SIM\*\* settings screen.

### Testing steps:

1. Flash a build containing this customization to a device.
2. Verify that extended error messages shown on the device when a reject code is sent by the network.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Hide CDMA mode selection

10/2/2018 • 2 minutes to read • [Edit Online](#)

For CDMA phones, partners can hide **CDMA** option in the network **Mode** selection drop-down that appears on the **Cellular & SIM** screen in **Settings**.

**Constraints:** None

This customization supports: **per-IMSI** value, **per-device** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="HideCDMAModeSelection"
    Description="Use to hide or show the 'CDMA' option in the network 'Mode' selection
    drop-down that appears in the cellular settings screen."
    Owner=""
    OwnerType="OEM">
```

```

<!-- Use for the per-IMSI case

<!-- Define the Targets -->
<Targets>
    <Target Id="">
        <TargetState>
            <Condition Name="" Value="" />
            <Condition Name="" Value="" />
        </TargetState>
    </Target>
</Targets>

<Static>
    <Settings Path="Multivariant">
        <Setting Name="Enable" Value="1" />
    </Settings>
    <Settings Path="AutoDataConfig">
        <Setting Name="Enable" Value="0" />
    </Settings>
</Static>

<!-- Specify the Variant -->
<Variant Name="">
    <TargetRefs>
        <TargetRef Id="" />
    </TargetRefs>

    <Settings Path="CellCore/PerIMSI/$(__IMSI)/CellUX">
        <!-- Hides or shows the 'CDMA' option in the network mode selection screen. Set to 0 or 'No' (to show)
        or set to 1 or 'Yes' (to hide). -->
        <Setting Name="Hide3GPP2ModeSelection" Value="" />
    </Settings>
</Variant>

-->

<!-- Use for the per-device case

<Static>
    <Settings Path="CellCore/PerDevice/CellUX">
        <!-- Hides or shows the 'CDMA' option in the network mode selection screen. Set to 0 or 'No' (to show)
        or set to 1 or 'Yes' (to hide). -->
        <Setting Name="Hide3GPP2ModeSelection" Value="" />
    </Settings>
</Static>

-->

</ImageCustomizations>
-->

```

1. Specify an `Owner`.
2. Determine if you need to use the **per-IMSI** or **per-device** setting.

For the **per-IMSI** case:

- a. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
  - b. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
3. Set the value for `Hide3GPP2ModeSelection` to one of the following:

| VALUE      | DESCRIPTION  |
|------------|--|
| 0 or 'No'  | Shows the <b>CDMA</b> option in the network <b>Mode</b> selection drop-down. |
| 1 or 'Yes' | Hides the <b>CDMA</b> option in the network <b>Mode</b> selection drop-down. |

The default for this setting is to show the \*\*CDMA\*\* option in the \*\*Mode\*\* selection drop-down that appears in the \*\*Cellular & SIM\*\* settings screen.

### Testing:

1. Flash the build containing this customization to a device.
2. Go to the **Cellular & SIM** screen in **Settings**.
3. Depending on the value that you set, verify whether the **CDMA** option in the network **Mode** selection drop-down is visible.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Hide Cellular & SIM Settings

10/2/2018 • 3 minutes to read • [Edit Online](#)

OEMs can hide certain user options for phones that appear in the **Cellular & SIM** screen in **Settings**.

These options include:

- For World mode: **Network Mode selection** drop-down
- For GSM: **Network Selection** drop-down
- For CDMA: **Network Type** drop-down

**Constraints:** None This customization supports: **per-IMSI** value, **per-device** value

## Instructions

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="CellularSettings"
    Description="Use to hide certain user options for phones that appear in the
    cellular+SIM settings screen."
    Owner=""
    OwnerType="OEM">

    <!-- Use for the per-IMSI case -->
    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>
    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>
        <Settings Path="CellCore/PerIMSI/$_IMSI/CellUX">
            <!-- Hides or shows the 'Network Mode selection' drop-down in the SIM settings screen for world
            mode phones.

            Set to 0 or 'No' (to show) or set to 1 or 'Yes' (to hide). -->
            <Setting Name="HideModeSelection" Value="" />

            <!-- Hides or shows the 'Network Selection' drop-down in the SIM settings screen for 3GPP or GSM
            phones.

            Set to 0 or 'No' (to show) or set to 1 or 'Yes' (to hide). -->
            <Setting Name="Hide3GPPNetworks" Value="" />
        </Settings>
    </Variant>
</ImageCustomizations>
```

```

<!-- Hides or shows the 'Network Type' drop-down in the SIM settings screen for 3GPP2 or CDMA phones.

    Set to 0 or 'No' (to show) or set to 1 or 'Yes' (to hide). -->
    <Setting Name="Hide3GPP2Selection" Value="" />
</Settings>
</Variant>

<!-- Use for the per-device case -->
<Static>
    <Settings Path="CellCore/PerDevice/CellUX">
        <!-- Hides or shows the 'Network Mode selection' drop-down in the SIM settings screen for world mode phones.

            Set to 0 or 'No' (to show) or set to 1 or 'Yes' (to hide). -->
            <Setting Name="HideModeSelection" Value="" />
        <!-- Hides or shows the 'Network Selection' drop-down in the SIM settings screen for 3GPP or GSM phones.

            Set to 0 or 'No' (to show) or set to 1 or 'Yes' (to hide). -->
            <Setting Name="Hide3GPPNetworks" Value="" />

        <!-- Hides or shows the 'Network Type' drop-down in the SIM settings screen for 3GPP2 or CDMA phones.

            Set to 0 or 'No' (to show) or set to 1 or 'Yes' (to hide). -->
            <Setting Name="Hide3GPP2Selection" Value="" />
        </Settings>
    </Static>
</ImageCustomizations>
```

2. Specify an `Owner`.

3. Determine if you need to use the **per-IMSI** or **per-device** setting.

For the **per-IMSI** case:

- Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
- Define settings for a **Variant**, which are applied if the associated target's conditions are met.

4. For World mode phones: Set the value for `HideModeSelection` to one of the following:

VALUE	DESCRIPTION
0 or 'No'	Shows the <b>Network Mode selection</b> drop-down in the <b>SIM</b> settings screen.
1 or 'Yes'	Hides the <b>Network Mode selection</b> drop-down in the <b>SIM</b> settings screen.

5. For 3GPP or GSM phones: Set the value for `Hide3GPPNetworks` to one of the following:

VALUE	DESCRIPTION
0 or 'No'	Shows the <b>Network Selection</b> drop-down in the <b>SIM</b> settings screen.
1 or 'Yes'	Hides the <b>Network Selection</b> drop-down in the <b>SIM</b> settings screen.

6. For 3GPP2 or CDMA phones: Set the value for `Hide3GPP2Selection` to one of the following:

VALUE	DESCRIPTION
0 or 'No'	Shows the <b>Network Type</b> drop-down in the <b>SIM</b> settings screen.
1 or 'Yes'	Hides the <b>Network Type</b> drop-down in the <b>SIM</b> settings screen.

## Testing

1. Flash the build containing this customization to a phone.
2. Go to the **Cellular & SIM** screen in **Settings**.
3. Verify that the user options are visible only if appropriate.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# LTE attach: GUID for user configured internet APN

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can set the OEMConnectionId that is used when creating the user-configured connection for internet from the **SIM** settings screen.

The value is a GUID in the string format "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX". It is used as the value for the OEMConnectionId field of the connection and it identifies the modem profile used for the LTE Attach. If this value is not set, the APN configuration entered by the user does not affect the LTE Attach GUID used by the device.

**Constraints:** None

This customization supports: **per-IMSI** value, **per-device** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>

<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="LTEAttachUserConfigGUID"
    Description="Use to set the OEMConnectionId used for the LTE attach profile in the
modem."
    Owner=""
    OwnerType="OEM">

<!-- Use for the per-IMSI case

    &lt!-- Define the Targets --&gt;
    &lt;Targets&gt;
        &lt;Target Id=""&gt;
            &lt;TargetState&gt;
                &lt;Condition Name="" Value="" /&gt;
                &lt;Condition Name="" Value="" /&gt;
            &lt;/TargetState&gt;
        &lt;/Target&gt;
    &lt;/Targets&gt;

    &lt;Static&gt;
        &lt;Settings Path="Multivariant"&gt;
            &lt;Setting Name="Enable" Value="1" /&gt;
        &lt;/Settings&gt;
        &lt;Settings Path="AutoDataConfig"&gt;
            &lt;Setting Name="Enable" Value="0" /&gt;
        &lt;/Settings&gt;
    &lt;/Static&gt;

    &lt!-- Specify the Variant --&gt;
    &lt;Variant Name=""&gt;
        &lt;TargetRefs&gt;
            &lt;TargetRef Id="" /&gt;
        &lt;/TargetRefs&gt;

        &lt;Settings Path="CellCore/PerIMSI/$(__IMSI)/CellUX"&gt;
            &lt;Setting Name="LTEAttachGUID" Value="" /&gt;
        &lt;/Settings&gt;
    &lt;/Variant&gt;
    --&gt;

    &lt!-- Use for the per-device case

    &lt;Static&gt;
        &lt;Settings Path="CellCore/PerDevice/CellUX"&gt;
            &lt;Setting Name="LTEAttachGUID" Value="" /&gt;
        &lt;/Settings&gt;
    &lt;/Static&gt;
    --&gt;
&lt;/ImageCustomizations&gt;</pre>

```

2. Specify an **Owner**.

3. Determine if you need to use the **per-IMSI** or **per-device** setting.

For the **per-IMSI** case:

- Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
- Define settings for a **Variant**, which are applied if the associated target's conditions are met.

4. Set the value for `LTEAttachGuid` to the OemConnectionId GUID used for the LTE attach profile in the modem. The value is a GUID in the string format "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX".

**Testing:**

Refer to the documentation provided by the modem vendor and work with your mobile operator to test this customization on their network.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# LTE attach: Mapping OEMConnectionId values to modem profiles

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can set the list of OEMConnectionId values that map to a LTE attach profile in the MBB driver. This list is used to specify which OEMConnectionIds require a detach/attach in order for changes to apply. If an OEMConnectionId is not included in this list and the LTE attach info is updated, it will not take effect until the device is rebooted.

**Constraints:** ImageTimeOnly

This customization supports: **per-device** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="LTEAttachProfileMap"
    Description="Use to set the list of OEMConnectionId values that map to a LTE attach
profile on the MBB driver side."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="CellCore/PerDevice/CellData/ModemProfiles">
            <Setting Name="LTEAttachGuids" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.
3. Set the value of `LTEAttachGuids` to the semicolon-separated list of OEMConnectionId values that map to a LTE attach profile on the MBB driver. OEMConnectionIds are GUIDs in the string format "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX".

## Testing:

Refer to the documentation provided by the modem vendor and work with your mobile operator to test this customization on their network.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Manual network selection timeout

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can change the default network selection timeout value. By default, the OS allows the device to register on the manually selected network for 60 seconds (or 1 minute) before it switches back to automatic mode.

**Constraints:** None

This customization supports: **per-device** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="ManualNetworkSelectionTimeout"
    Description="Use to change the default network selection timeout value."
    Owner=""
    OwnerType="OEM">

    <!-- Use for the per-device case

    <Static>
        <Settings Path="CellCore/PerDevice/General">
            <Setting Name="ManualNetworkSelectionTimeout" Value="" />
        </Settings>
    </Static>

    -->

</ImageCustomizations>
```

2. Specify an `Owner`.
3. Set the `ManualNetworkSelectionTimeout``Value` to a desired timeout value. The range of the value can be from 1-420 seconds. For example, to change the value to 120 seconds (or 2 minutes), you must set the value to `0x78`.

This value is the amount of time that the OS will wait for the modem to register on the manually selected network. If the time lapses and the modem was not able to register on the network that was manually selected by the user, the OS will either:

- Switch back to the automatic network selection mode if **Permanent automatic mode** is enabled and after the user has manually selected a network or the modem was turned on.
- Display a dialog that notifies the user that the device was unable to connect to the manually selected network after the device was turned on or after airplane mode was turned off.

## Testing steps:

### Important

To fully test this customization, you must work with your mobile operator partner to perform Steps 2 and 3 in the following procedure.

1. Flash the build that contains this customization to a device that has a UICC.
2. Ensure that the device is out of range from the home network.

3. Set the device to the manual network mode by selecting **manual** mode under **Network selection** in the **Settings > Cellular & SIM** screen.
4. While the device attempts to connect to the manually selected network, verify that the OS waits for the amount of time that you specified for `ManualNetworkSelectionTimeout` before it switches back to the automatic network selection mode, or displays a message that indicates that the device was unable to connect to the manually selected network.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Maximum number of PDP contexts

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can set different maximum values for the number of PDP contexts for the device for 3GPP network if required by their mobile operator.

By default, the OS enforces a maximum of four (4) simultaneous packet data protocol (PDP) contexts for 3GPP connections, and one (1) PDP context for 3GPP2 connections.

The same maximums apply for both roaming and non-roaming scenarios. This maximum does not include packet contexts used internally by the modem.

## Constraints:

1. The setting `MaxNumberOfPDPContexts` is applicable for 3GPP networks only. It does not apply to 3GPP2 network.
2. This customization supports: **per-IMSI** value, **per-device** value.

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>

<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="MaxNumberOfPDPContexts"
    Description="Use to set the maximum number of concurrent packet contexts for the
    home carrier's 3GPP network"
    Owner=""
    OwnerType="OEM">

    <!-- Use for the per-IMSI case

        <!-- Define the Targets -->
        <Targets>
            <Target Id="">
                <TargetState>
                    <Condition Name="" Value="" />
                    <Condition Name="" Value="" />
                </TargetState>
            </Target>
        </Targets>

        <Static>
            <Settings Path="Multivariant">
                <Setting Name="Enable" Value="1" />
            </Settings>
            <Settings Path="AutoDataConfig">
                <Setting Name="Enable" Value="0" />
            </Settings>
        </Static>

        <!-- Specify the Variant -->
        <Variant Name="">
            <TargetRefs>
                <TargetRef Id="" />
            </TargetRefs>

            <Settings Path="CellCore/PerIMSI/$(__IMSI)/CellData">
                <Setting Name="MaxNumberOfPDPContexts" Value="" />
            </Settings>
        </Variant>
    -->

    <!-- Use for the per-device case

        <Static>
            <Settings Path="CellCore/PerDevice/CellData">
                <Setting Name="MaxNumberOfPDPContexts" Value="" />
            </Settings>
        </Static>
    -->

```

```

</ImageCustomizations>
```

```

1. Specify an `Owner`.
2. Determine if you need to use the **per-IMSI** or **per-device** setting.

For the **per-IMSI** case:

- a. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
  - b. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
3. Set the value for `MaxNumberOfPDPContexts` as required by the mobile operator. You can specify a value between 1 through 4 (inclusive), or 0x1 through 0x4 (hexadecimal).

**Testing:**

Work with your mobile operator partner to test this customization on their network.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Permanent automatic mode

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can enable permanent automatic mode for mobile networks that require the cellular settings to revert to automatic network selection after the user has manually selected another network when roaming or out of range of the home network.

**Constraints:** None This customization supports: **per-IMSI** value, **per-device** value

## Instructions

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="PermanentAutomaticMode"
    Description="Use to enable permanent automatic mode."
    Owner=""
    OwnerType="OEM">

    <!-- Use for the per-IMSI case -->
    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="CellCore/PerIMSI/$(__IMSI)/General">
            <Setting Name="AvoidStayingInManualSelection" Value="" />
        </Settings>
    </Variant>

    <!-- Use for the per-device case -->
    <Static>
        <Settings Path="CellCore/PerDevice/General">
            <Setting Name="AvoidStayingInManualSelection" Value="" />
        </Settings>
    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Determine if you need to use the **per-IMSI** or **per-device** setting.

For the **per-IMSI** case:

a. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.

b. Define settings for a **Variant**, which are applied if the associated target's conditions are met.

4. Set the `AvoidStayingInManualSelection``Value` to either of the following:

VALUE	DESCRIPTION
1	Enable permanent automatic mode.
0	Disable permanent automatic mode. The cellular settings for network selection remain in manual mode.

## Testing

1. Flash the build containing this customization to a device that has a UICC.

### NOTE

To fully test this customization, work with your mobile operator partner. The device needs to be out of range of the home network so that the user can select **manual** mode under **Network selection** in the **Settings > Cellular & SIM** screen.

2. When the device is no longer roaming or is in range of the home network, verify that the **Network selection** mode changed to **automatic** without requiring user action.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Preferred data provider list

10/2/2018 • 2 minutes to read • [Edit Online](#)

For mobile operators that require it, OEMs can set a list of MCC/MNC pairs for the purchase order (PO) carrier or primary operator so that it can be set as the default data line for phones that have a dual SIM.

When the PO SIM is inserted into the phone, the OS picks the PO SIM as the data line and shows a notification to the user that the SIM has been selected for internet data. If two PO SIMs are inserted, the OS will choose the first PO SIM that was detected as the default data line and the mobile operator action required dialogue (ARD) is shown. If two non-PO SIMs are inserted, the user is prompted to choose the SIM to use as the default data line.

## Note

OEMs should not set this customization unless required by the mobile operator.

**Constraints:** None

**Instructions:**

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>

<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="PreferredDataProviderList"
    Description="Use to specify the list of preferred mobile operators' MCC and MNC
                information to use for data connections."
    Owner=""
    OwnerType="OEM">

    <Static>
        <Settings Path="CellCore/PerDevice/General">
            <Setting Name="PreferredDataProviderList" Value="" />
        </Settings>
    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.
3. To enumerate the MCC/MNC value pairs to use for data connections, set the value for `PreferredDataProviderList`. The value must be a comma-separated list of preferred MCC:MNC values. For example, the value can be `301:026,310:030` and so on.

**Testing:**

1. Work with your mobile operator to obtain the list of preferred MCC and MNC values for data connections.
2. Flash the build containing this customization to a dual SIM phone.
3. Insert the PO SIM into the phone. Verify that the OS picks the PO SIM as the default data line and shows a notification that the SIM has been selected for Internet data.
4. Insert two PO SIMs. Verify that the OS chooses the first PO SIM that was detected as the default data line and the mobile operator action required dialogue (ARD) is shown.
5. Insert two non-PO SIMs. Verify that you can see a prompt to choose the SIM to use as the default data line.

6. Verify that you can change the default SIM by going to the **Cellular+SIM** settings screen and selecting a different SIM to use as the default data line.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Remove cellular functionality from the device

10/2/2018 • 2 minutes to read • [Edit Online](#)

If your mobile device does not support a cellular radio or will not be connected to a cellular network, you can remove all cellular-related functionality from the device's user interface by adding the WIFI\_FEATURE\_PACK feature entry in your OEMInput.xml file. This feature replaces the WEH\_WIFIONLY feature that you previously used in earlier versions of the mobile OS.

The WIFI\_FEATURE\_PACK package reduces memory usage and improves the user experience by removing the non-functioning cellular-related tiles, icons, and settings. Wi-Fi features will continue to work and airplane mode will also work.

## Instructions:

### To create an update package for an existing mobile device using Windows 10 Mobile

1. Update your OS image using Windows 10 Mobile as your new base image.
2. Remove the WEH\_WIFIONLY Microsoft Update package if you are upgrading from Windows Embedded Handheld 8.1.
3. Add the WIFI\_FEATURE\_PACK as a BSP update.
4. Test, sign, and submit the update.

### To create a new Windows 10 Mobile image without cellular functionality

1. Locate the OEMInput.xml file that you are using to define your image.
2. Find the **Features** section, and within the **Microsoft** child element, review the **Feature** elements.
3. Add a <Feature>WIFI\_FEATURE\_PACK</Feature> entry in your OEMInput.xml file.

```
<Features>
  <Microsoft>
    <Feature>WIFI_FEATURE_PACK</Feature>
  </Microsoft>
</Features>
```

For more information about other features you can include in your image, see [Optional features for building images](#).

4. Save the updated OEMInput.xml file and build your mobile OS image.
5. Verify that the new image doesn't contain cellular-related tiles, icons, and settings. Also verify that Wi-Fi features work and airplane mode functions correctly.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Roaming filter

10/2/2018 • 5 minutes to read • [Edit Online](#)

Partners can add roaming filters that determine when the device appears to be roaming, based on the network the device is currently connected to. With roaming filters enabled, connections on other companies' specified networks are not treated as roaming.

## Constraints: Atomic

This customization supports: **per-IMSI** value, **per-device** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="RoamingFilter"
    Description="Use to add 3GPP or 3GPP2 filters that determine when the phone appears
    to be roaming based on the network the phone is currently connected to."
    Owner=""
    OwnerType="OEM">

    <!-- Use for the per-IMSI case

    &lt;!-- Define the Targets --&gt;
    &lt;Targets&gt;
        &lt;Target Id=""&gt;
            &lt;TargetState&gt;
                &lt;Condition Name="" Value="" /&gt;
                &lt;Condition Name="" Value="" /&gt;
            &lt;/TargetState&gt;
        &lt;/Target&gt;
    &lt;/Targets&gt;

    &lt;Static&gt;
        &lt;Settings Path="Multivariant"&gt;
            &lt;Setting Name="Enable" Value="1" /&gt;
        &lt;/Settings&gt;
        &lt;Settings Path="AutoDataConfig"&gt;
            &lt;Setting Name="Enable" Value="0" /&gt;
        &lt;/Settings&gt;
    &lt;/Static&gt;

    &lt;!-- Specify the Variant --&gt;
    &lt;Variant Name=""&gt;
        &lt;TargetRefs&gt;
            &lt;TargetRef Id="" /&gt;
        &lt;/TargetRefs&gt;

        &lt;!-- Define for 3GPP. All these settings must be configured at the same time. --&gt;
        &lt;Settings Path="CellCore/PerIMSI/$(__IMSI)/General/atomicRoamingTableSettings3GPP"&gt;
            &lt;Setting Name="RoamingTables/3GPPRoamingTables/Enabled" Value="" /&gt;

            &lt;Setting Name="RoamingTables/3GPPRoamingTables/TargetImsi/$(SerialNumber)" Value="" /&gt;
            &lt;Setting Name="RoamingTables/3GPPRoamingTables/TargetImsi/$(SerialNumber)" Value="" /&gt;
            &lt;Setting Name="RoamingTables/3GPPRoamingTables/TargetImsi/$(SerialNumber)" Value="" /&gt;

            &lt;Setting Name="RoamingTables/3GPPRoamingTables/Exceptions/$(SerialNumber)" Value="" /&gt;
            &lt;Setting Name="RoamingTables/3GPPRoamingTables/Exceptions/$(SerialNumber)" Value="" /&gt;
            &lt;Setting Name="RoamingTables/3GPPRoamingTables/Exceptions/$(SerialNumber)" Value="" /&gt;</pre>
```

```

<Setting Name="RoamingTables/3GPPRoamingTables/HomePLMN/${SerialNumber}" Value="" />
<Setting Name="RoamingTables/3GPPRoamingTables/HomePLMN/${SerialNumber}" Value="" />
<Setting Name="RoamingTables/3GPPRoamingTables/HomePLMN/${SerialNumber}" Value="" />
</Settings>

<!-- Define for 3GPP2. All these settings must be configured at the same time. -->
<Settings Path="CellCore/PerIMSI/$_IMSI/General/atomicRoamingTableSettings3GPP2">
    <Setting Name="RoamingTables/3GPP2RoamingTables/Enabled" Value="" />

    <Setting Name="RoamingTables/3GPP2RoamingTables/Home/${SerialNumber}" Value="" />
    <Setting Name="RoamingTables/3GPP2RoamingTables/Home/${SerialNumber}" Value="" />
    <Setting Name="RoamingTables/3GPP2RoamingTables/Home/${SerialNumber}" Value="" />

    <Setting Name="RoamingTables/3GPP2RoamingTables/Roaming/${SerialNumber}" Value="" />
    <Setting Name="RoamingTables/3GPP2RoamingTables/Roaming/${SerialNumber}" Value="" />
    <Setting Name="RoamingTables/3GPP2RoamingTables/Roaming/${SerialNumber}" Value="" />
</Settings>

</Variant>

-->

<!-- Use for the per-device case

<Static>

    <!-- Define for 3GPP. All these settings must be configured at the same time. -->
    <Settings Path="Cellcore/PerDevice/General/atomicRoamingTableSettings3GPP">
        <Setting Name="RoamingTables/3GPPRoamingTables/Enabled" Value="" />

        <Setting Name="RoamingTables/3GPPRoamingTables/TargetImsi/${SerialNumber}" Value="" />
        <Setting Name="RoamingTables/3GPPRoamingTables/TargetImsi/${SerialNumber}" Value="" />
        <Setting Name="RoamingTables/3GPPRoamingTables/TargetImsi/${SerialNumber}" Value="" />

        <Setting Name="RoamingTables/3GPPRoamingTables/Exceptions/${SerialNumber}" Value="" />
        <Setting Name="RoamingTables/3GPPRoamingTables/Exceptions/${SerialNumber}" Value="" />
        <Setting Name="RoamingTables/3GPPRoamingTables/Exceptions/${SerialNumber}" Value="" />

        <Setting Name="RoamingTables/3GPPRoamingTables/HomePLMN/${SerialNumber}" Value="" />
        <Setting Name="RoamingTables/3GPPRoamingTables/HomePLMN/${SerialNumber}" Value="" />
        <Setting Name="RoamingTables/3GPPRoamingTables/HomePLMN/${SerialNumber}" Value="" />
    </Settings>

    <!-- Define for 3GPP2. All these settings must be configured at the same time. -->
    <Settings Path="Cellcore/PerDevice/General/atomicRoamingTableSettings3GPP2">
        <Setting Name="RoamingTables/3GPP2RoamingTables/Enabled" Value="" />

        <Setting Name="RoamingTables/3GPP2RoamingTables/Home/${SerialNumber}" Value="" />
        <Setting Name="RoamingTables/3GPP2RoamingTables/Home/${SerialNumber}" Value="" />
        <Setting Name="RoamingTables/3GPP2RoamingTables/Home/${SerialNumber}" Value="" />

        <Setting Name="RoamingTables/3GPP2RoamingTables/Roaming/${SerialNumber}" Value="" />
        <Setting Name="RoamingTables/3GPP2RoamingTables/Roaming/${SerialNumber}" Value="" />
        <Setting Name="RoamingTables/3GPP2RoamingTables/Roaming/${SerialNumber}" Value="" />
    </Settings>

</Static>

-->

</ImageCustomizations>
```

2. Specify an **Owner**.

3. Determine if you need to use the **per-IMSI** or **per-device** setting.

For the **per-IMSI** case:

- a. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
  - b. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
4. Use the correct settings for 3GPP or 3GPP2. You must work with your mobile operator to obtain the correct values specific to the mobile operator.

The settings group is atomic so you must configure all the settings at the same time to correctly configure roaming filters.

#### **IMPORTANT**

When specifying the `$(serialNumber)`, make sure that the order is absolutely sequential within the different lists. For example: 000, 001, 002..., or 001, 002, 003..., and so on.

## **3GPP**

- a. To enable the 3GPP filter, set `RoamingTables/3GPPRoamingTables/Enabled` to 1 or Yes. Setting the value to 0 or No disables the 3GPP filter.
- b. The `RoamingTables/3GPPRoamingTables/TargetImsi/$(SerialNumber)` setting defines all the possible Mobile Country Code (MCC)-Mobile Network Code (MNC) pairs that can be included in the IMSI encoded on the mobile operator's SIM cards. If the MCC-MNC value on the SIM in the device does not match one of these pairs, the SIM is recognized as belonging to another mobile operator, and the roaming filter is not enabled. The values in this registry key are specific to the mobile operator.

Replace `$(SerialNumber)` to correspond to the 3-digit serial number, from 000 through 999, represented as a string. For each serial number that you defined, set the value to a string representing the **MCC,MNC**, such as **410,510** to represent an MCC of 410 and MNC of 510, for example.

Add as many `RoamingTables/3GPPRoamingTables/TargetImsi/$(SerialNumber)` settings and values as you need.

- c. The `RoamingTables/3GPPRoamingTables/Exceptions/$(SerialNumber)` setting defines the MCC-MNC values for networks on which the device is roaming within the group of home codes that you define.

Replace `$(SerialNumber)` to correspond to the 3-digit serial number, from 000 through 999, represented as a string. For each serial number that you defined, set the value to a string representing the **MCC,MNC**, such as **410,510** to represent an MCC of 410 and MNC of 510, for example.

Add as many `RoamingTables/3GPPRoamingTables/Exceptions/$(SerialNumber)` settings and values as you need.

As with the rest of these settings, the exact MNC and MCC values are mobile operator-specific.

- d. The `RoamingTables/3GPPRoamingTables/HomePLMN/$(SerialNumber)` setting defines the network codes where the device is not deemed roaming. These settings can include just an MCC, or they can consist of a MCC-MNC pair. These settings are specific to the mobile operator.

Replace `$(SerialNumber)` to correspond to the 3-digit serial number, from 000 through 999, represented as a string. For each serial number that you defined, set the value to a string representing the **MCC,MNC**, such as **410,510** to represent an MCC of 410 and MNC of 510, for example. Alternatively, you can also set the Value to a string that represents just the **MCC**, such as **460** for MCC of 460 and all networks of that country.

Add as many `RoamingTables/3GPPRoamingTables/HomePLMN/$(SerialNumber)` settings and values as you need.

## 3GPP2

- a. To enable the 3GPP2 filter, set `RoamingTables/3GPP2RoamingTables/Enabled` to 1 or Yes. Setting the value to 0 or No disables the 3GPP2 filter.
- b. The `RoamingTables/3GPP2RoamingTables/Home/$(SerialNumber)` setting defines the network codes where the device is not deemed roaming. These settings are specific to the mobile operator.

Replace `$(SerialNumber)` to correspond to the 3-digit serial number, from 000 through 999, represented as a string. For each serial number that you defined, set the value to a DWORD representing the non-roaming indicator.

Add as many `RoamingTables/3GPP2RoamingTables/Home/$(SerialNumber)` settings and values as you need until you have every valid code added.

- c. The `RoamingTables/3GPP2RoamingTables/Roaming/$(SerialNumber)` setting defines the values for networks on which the device is deemed roaming.

Replace `$(SerialNumber)` to correspond to the 3-digit serial number, from 000 through 999, represented as a string. For each serial number that you defined, set the value to a DWORD representing the roaming indicator.

Add as many `RoamingTables/3GPP2RoamingTables/Roaming/$(SerialNumber)` settings and values as you need until you have every valid code added.

### Testing steps:

Work with your mobile operator to test this customization on their network.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Customizations for Wi-Fi settings and connectivity

10/2/2018 • 2 minutes to read • [Edit Online](#)

Describes the customizations that you can configure for Wi-Fi settings and Wi-Fi connectivity on the mobile device.

## In this section

TOPIC	DESCRIPTION
<a href="#">Authentication for Wi-Fi hotspot settings</a>	Set a list of captive portal SSIDs for which the browser should not be launched automatically for Wi-Fi authentication.
<a href="#">Cellular data fallback when in limited Wi-Fi connectivity</a>	OEMs can change the default behavior for the device when Wi-Fi connectivity becomes limited.
<a href="#">Change Wi-Fi to WLAN</a>	To meet regulatory requirements and/or meet mobile operator requirements for some markets, partners can replace the string Wi-Fi with the generic term WLAN. Enabling this customization changes all Wi-Fi strings to WLAN.
<a href="#">Connecting to open Wi-Fi hotspots in Windows 10</a>	Partners can change the default settings for detecting and auto-connecting to Wi-Fi hotspots.
<a href="#">Enable static IP</a>	To facilitate Wi-Fi certification tests, OEMs can enable a screen from the Wi-Fi settings screen that provides UI elements that allow you to specify a static IP address, gateway address, and DNS server address.
<a href="#">Limited connectivity status</a>	By default, when the device is connected to a Wi-Fi access point (AP), it does not show a No Internet access status message below the AP name. Partners may choose to override this default behavior and show the status message when a device is connected to a Wi-Fi access point.
<a href="#">Wi-Fi always on, always connected</a>	Partners can modify AOAC behavior and UX for non-AOAC mode devices.
<a href="#">Wi-Fi calling errors</a>	OEMs can customize the mobile device to configure settings related to Wi-Fi calling errors.
<a href="#">Wi-Fi calling operator name</a>	OEMs can customize the display name for the mobile operator when the device is using Wi-Fi calling.
<a href="#">Wi-Fi icon and notifications</a>	Partners can configure settings related to the Wi-Fi icon.

## Related topics

[Customizations for Cellular connectivity](#)

[Prepare for Windows mobile development](#)

## Customization answer file overview

# Authentication for Wi-Fi hotspot settings

10/2/2018 • 2 minutes to read • [Edit Online](#)

When mobile devices connect to a Wi-Fi hotspot that uses a captive portal, the web browser is automatically opened so that the user can sign in. Until this authentication process is completed, the Wi-Fi hotspot connection is not available for applications and services running on the device. As a result, applications provided by mobile operators or third party vendors to authenticate the Wi-Fi hotspot settings will not work.

To suppress the launch of the browser and enable this type of application to run as expected, OEMs must register the SSID of one or more networks. These networks will be available immediately when the connection is established.

**Constraints:** None

**Instructions:**

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>

<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="HijackedIgnoreList"
    Description="Use specify a list of captive portal SSIDs for which browser should
not be launched automatically."
    Owner=""
    OwnerType="OEM">

    <Static>
        <Settings Path="WiFi/Config">
            <Setting Name="HijackedIgnoreList" Value="" />
        </Settings>
    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.
3. Set the `HijackedIgnoreList` value to the set list of captive portal SSIDs for which the browser should not be launched automatically. For example, `ContosoWiFi;FabrikamWiFi;ContosoFabrikamWiFi` and so on.

**Testing steps:**

1. Flash the build containing this customization to a device.
2. Connect to a Wi-Fi network that uses one of the registered SSIDs.
3. Verify the browser does not launch and the network is available immediately.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Cellular data fallback when in limited Wi-Fi connectivity

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can change the default behavior for the device when Wi-Fi connectivity becomes limited.

By default, if the device is connected to a Wi-Fi network and the data connection to a site is unsuccessful due to limited Wi-Fi connectivity, the device will complete the connection to the site using available cellular data networks (when possible) to provide an optimal user experience.

OEMs can change this default behavior so that the device does not use cellular data when Wi-Fi connectivity becomes limited. When the customization is enabled, a user option to use or not use cellular data for limited Wi-Fi connectivity becomes visible in the **Cellular & SIM** settings screen. This option is automatically set to **don't use cellular data** when the customization is enabled.

## Note

Changing the default behavior may negatively impact the user experience.

**Constraints:** None

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="CellularFailover"
    Description="Use to allow or disallow cellular data fallback when in limited Wi-Fi
    connectivity."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Cellcore/PerDevice/CellData">
            <Setting Name="CellularFailOver" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set the value of `CellularFailover` to one of the following:

VALUE	DESCRIPTION
0 or 'Failover is not allowed'	Disables cellular data fallback when in limited Wi-Fi connectivity. This also sets the <b>For limited Wi-Fi connectivity</b> option in the <b>Cellular &amp; SIM</b> settings screen to <b>don't use cellular data</b> .

VALUE	DESCRIPTION
1 or 'Failover is allowed'	Enables cellular data fallback when in limited Wi-Fi connectivity.

**Testing:**

1. Flash the build containing this customization to a device with a UICC.
2. If you set `CellularFailover` to 0 or 'Failover is not allowed', navigate to the **Cellular & SIM** settings screen and verify that the **don't use cellular data** option is chosen as the default under **For limited Wi-Fi connectivity**.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Change Wi-Fi to WLAN

10/2/2018 • 2 minutes to read • [Edit Online](#)

To meet regulatory requirements and/or meet mobile operator requirements for some markets, partners can replace the string **Wi-Fi** with the generic term **WLAN**. Enabling this customization changes all **Wi-Fi** strings to **WLAN**.

**Constraints:** ImageTimeOnly

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="WiFiToWLAN"
    Description="Use To replace the 'Wi-Fi' strings in the phone UI to 'WLAN' to meet
    mobile operator or regulatory requirements."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="WiFi/FirstBoot">
            <Setting Name="WiFiToWLAN" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set the value of `WiFiToWLAN` to one of the following:

VALUE	DESCRIPTION
1 or 'Enabled'	Use to enable or replace all "Wi-Fi" strings with "WLAN".
0 or 'Disabled'	Use to disable the customization. This is the default behavior.

## Testing steps:

1. Flash a build containing this customization to a phone.
2. Go to the **Settings** screen and verify that **WLAN** now shows up instead of **Wi-Fi**.
3. Tap **WLAN**, and verify that "Wi-Fi" does not appear in the WLAN setting screen.

All other "Wi-Fi" strings on the phone should now show "WLAN".

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Wi-Fi hotspots

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can change the default settings for detecting Wi-Fi hotspots.

Windows 10 automatically connects users to Wi-Fi so they can get online quickly in more places. It can connect them to open Wi-Fi hotspots that it knows about through crowdsourcing.

## How it works

Users choose the settings for automatically connecting to suggested open hotspots by going to Settings > Network & Internet > Wi-Fi on a Windows 10 PC or a phone with Windows 10 Mobile in Settings > Network & wireless > Wi-Fi > Additional settings. To use this feature, customers will need to be signed in with their Microsoft account on your Windows 10 PC or mobile device. (Note that this feature isn't available in all countries or regions.)

Windows 10 learns about open Wi-Fi hotspots a Windows PC or Windows phone connects to by collecting information about the network, like whether the open Wi-Fi network has a high-quality Internet connection. By using that information from your device and from other Windows customers' devices, we build a database of these high-quality networks. When you're in range of one of these Wi-Fi hotspots, you automatically get connected to it.

**Constraints:** ImageTimeOnly

### Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="WiFiConnections"
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="WiFi/FirstBoot">
            <Setting Name="AutoConnectAllowed" Value="" />
            <Setting Name="DefaultAutoConnectState" Value="" />
            <Setting Name="DefaultWiFiSharingState" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set the value of `AutoConnectAllowed` to one of the following values:

VALUE	DESCRIPTION
1 or 'Enabled'	Use to enable Wi-Fi detection. When enabled, users can opt-in to Wi-Fi detection.  This is the default OS value.

VALUE	DESCRIPTION
0 or 'Disabled'	Use to disable Wi-Fi detection.

4. Set the value of `DefaultAutoConnectState` to one of the following values:

VALUE	DESCRIPTION
1 or 'Enabled'	Sets the checkbox for <b>Automatically connect to Wi-Fi networks and accept terms for me</b> during initial phone setup.  This is the default OS value.
0 or 'Disabled'	Clears the checkbox for <b>Automatically connect to Wi-Fi networks and accept terms for me</b> during initial phone setup.

5. Set the value of `DefaultWiFiSharingState` to one of the following values:

VALUE	DESCRIPTION
1 or 'Enabled'	Sets the checkbox for <b>Allow me to exchange Wi-Fi network access with my contacts</b> during initial phone setup.  This is the default OS value.
0 or 'Disabled'	Clears the checkbox for <b>Allow me to exchange Wi-Fi network access with my contacts</b> during initial phone setup.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Enable static IP

10/2/2018 • 2 minutes to read • [Edit Online](#)

To facilitate Wi-Fi certification tests, OEMs can enable a screen from the Wi-Fi settings screen that provides UI elements that allow you to specify a static IP address, gateway address, and DNS server address.

To enable the **Static IP** UI, set the value of the `EnableStaticIP` setting to 1. If the setting is not set, or is set to any value other than 1, the static IP UI is not enabled. When enabled, the Wi-Fi **Static IP** UI button appears directly below the **Advanced** button in the Wi-Fi settings screen.

## Warning

The static IP UI must only be used for certification purposes and not for production or retail devices.

**Constraints:** None

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="EnableStaticIP"
    Description="Use to show the static IP settings in the advanced Wi-Fi settings
screen.
    This customization is for testing purposes only and should not be set
in
    production or retail images."
    Owner=""
    OwnerType="OEM">

<Static>
    <Settings Path="WiFi/Config">
        <Setting Name="EnableStaticIP" Value="" />
    </Settings>
</Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set the value of `EnableStaticIP` to one of the following:

VALUE	DESCRIPTION
1	Use to show the <b>Static IP</b> settings under <b>Settings</b> > <b>Wi-Fi</b> > <b>Static IP</b> .
0	Use to disable the customization.

## Testing steps:

1. Flash a build containing this customization to a device.

2. Go to the **Settings > Wi-Fi** screen and connect to a Wi-Fi network.
3. From the Wi-Fi settings screen, verify that you can see the **Static IP** setting.
4. Tap **Static IP** and configure your IP settings.
5. Reboot the device.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Limited connectivity status

10/2/2018 • 2 minutes to read • [Edit Online](#)

By default, when the device is connected to a Wi-Fi access point (AP), it does not show a **No Internet access** status message below the AP name. Partners may choose to override this default behavior and show the status message when a device is connected to a Wi-Fi access point.

## WARNING

The message may cause user confusion because it is shown whenever a proxy is used, as hotspot plug-in probes and data do not go through a proxy.

**Constraints:** None

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="LimitedConnectivityStatus"
    Description="Use to show the \"No Internet access\" status in the Wi-Fi settings
page when connectivity is limited."
    Owner=""
    OwnerType="OEM">
    <Static>
        <Settings Path="WiFi/Config">
            <Setting Name="PublishLimitedConnectivity" Value="" />
        </Settings>
    </Static>
</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set `PublishLimitedConnectivity` to one of the following values:

VALUE	DESCRIPTION
0 or 'Disabled'	Do not show the <b>No Internet access</b> status message below the AP name. This is the default OS behavior.
1 or 'Enabled'	Show the <b>No Internet access</b> status message below the AP name.

## Testing steps:

1. Flash the build that contains this customization to a device.
2. Connect to a Wi-Fi access point.
3. Depending on the value you set for `PublishLimitedConnectivity`, verify whether the **No Internet access** status message is shown below the AP name.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Wi-Fi always on, always connected

10/2/2018 • 3 minutes to read • [Edit Online](#)

Partners can modify AOAC behavior and UX for non-AOAC mode devices.

Partners can use the **LowPowerSupported** and **AlwaysOnAlwaysConnected** settings to modify AOAC behavior and UX. The device's supported AOAC mode is determined by a combination of the chipset, IHV driver, and the **LowPowerSupported** setting.

- **LowPowerSupported** – This setting specifies that the IHV driver partially supports AOAC. This setting must only be used if the IHV driver supports a transition from the D0 state to the D2 state and certain low power features.
- **AlwaysOnAlwaysConnected** – This setting enables partners to specify whether Wi-Fi should remain on when the screen times out. By default, this setting is disabled. Partners should note that this setting does not apply to devices that support partial or full AOAC. In that case, Wi-Fi always remains on and in the lower power state when the screen is idle. Also note that this setting controls the default state of the checkbox **Keep Wi-Fi on when the screen times out** in the **Wi-Fi > manage** settings screen. This checkbox is only visible for non-AOAC mode devices.

**Constraints:** ImageTimeOnly for LowPowerSupported

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="WiFiAOAC"
    Description="Use to configure the Wi-Fi driver to support transition from a D0
state to a D2 state and to specify whether
    Wi-Fi should stay on when the screen times out."
    Owner=""
    OwnerType="OEM">

    <Static>

        <!-- This setting is ImageTimeOnly. Specifies if the Wi-Fi driver supports D0 to D2 transitioning.
            Enable this to indicate 'partial' AOAC state.
        <Settings Path="WiFi/FirstBoot">
            <!-- Set to 0 or 'Disabled' (to disable), or set to 1 or 'Enabled' (to enable). -->
            <Setting Name="LowPowerSupported" Value="" />
        </Settings>
        -->

        <!-- Configures the Wi-Fi radios to always stay on even after the screen times out. This applies to
non-AOAC devices only.
        <Settings Path="WiFi/Config">
            <!-- Set to 0 or 'Disabled' (to disable), or set to 1 or 'Enabled' (to enable). -->
            <Setting Name="AlwaysOnAlwaysConnected" Value="" />
        </Settings>
        -->

    </Static>

</ImageCustomizations>
```

- Specify an `Owner`.
- To specify whether the Wi-Fi driver supports transitions from the D0 state to the D2 state and the required low power features, configure the value for `LowPowerSupported` to one of the following values:

VALUE	DESCRIPTION
0 or 'Disabled'	IHV driver does not support transitions from the D0 state to the D2 state and the required low power features.
1 or 'Enabled'	IHV driver supports transitions from the D0 state to the D2 state and the required low power features.

- To specify whether Wi-Fi should remain on when the screen times out, configure the value for `AlwaysOnAlwaysConnected` to one of the following values:

VALUE	DESCRIPTION
0 or 'Disabled'	Disables Wi-Fi from always being on when the screen times out. The <b>Keep Wi-Fi on when the screen times out</b> in the <b>Settings &gt; Wi-Fi &gt; manage</b> screen is turned off.
1 or 'Enabled'	Enables Wi-Fi to always be on by default when the screen times out. The <b>Keep Wi-Fi on when the screen times out</b> in the <b>Settings &gt; Wi-Fi &gt; manage</b> screen is turned on.

## Testing:

- Flash the build containing this customization to a device that is connected to a Wi-Fi network.
- If your Wi-Fi driver supports a D0 to D2 state transition and you enabled `LowPowerSupported`, verify that the device transitions from a D0 state to a D2 state.
- If you have a non-AOAC device and you configured the `AlwaysOnAlwaysConnected` setting, verify whether Wi-Fi remains on when the screen times out. Navigate to the **Wi-Fi > manage** settings screen and verify that the **Keep Wi-Fi on when the screen times out** setting is set according to the value that you specified for `AlwaysOnAlwaysConnected`.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Wi-Fi calling errors

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can customize the mobile device to configure settings related to Wi-Fi calling errors, including:

- Show an error message when a Wi-Fi calling error is reported by the modem.
- Show a specific error message based on operator requirements.
- Customize the generic error string when a Wi-Fi calling error happens.

**Constraints:** None

This customization supports: **per-IMSI** value, **per-device** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="WiFiCallingErrors"
    Description="Use to customize the Wi-Fi calling error settings."
    Owner=""
    OwnerType="OEM">

    <!-- Use for the per-IMSI case

        <!-- Define the Targets -->
        <Targets>
            <Target Id="">
                <TargetState>
                    <Condition Name="" Value="" />
                    <Condition Name="" Value="" />
                </TargetState>
            </Target>
        </Targets>

        <Static>
            <Settings Path="Multivariant">
                <Setting Name="Enable" Value="1" />
            </Settings>
            <Settings Path="AutoDataConfig">
                <Setting Name="Enable" Value="0" />
            </Settings>
        </Static>

        <!-- Specify the Variant -->
        <Variant Name="">
            <TargetRefs>
                <TargetRef Id="" />
            </TargetRefs>

            <Settings Path="CellCore/PerIMSI/$(__IMSI)/CellUX">
                <Setting Name="ShowWifiCallingError" Value="" />
                <Setting Name="ShowSpecificWifiCallingError" Value="" />
                <Setting Name="GenericWifiCallingErrorMessage" Value="" />
            </Settings>
        </Variant>
    <!-->

    <!-- Use for the per-device case

        <Static>
            <Settings Path="CellCore/PerDevice/CellUX">
                <Setting Name="ShowWifiCallingError" Value="" />
                <Setting Name="ShowSpecificWifiCallingError" Value="" />
                <Setting Name="GenericWifiCallingErrorMessage" Value="" />
            </Settings>
        </Static>
    <!-->

</ImageCustomizations>

```

2. Specify an `Owner`.

3. Determine if you need to use the **per-IMSI** or **per-device** setting.

For the **per-IMSI** case:

- Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.

- b. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
4. To show the Wi-Fi calling error message in the mobile device UI, set `ShowWifiCallingError` to one of the following values:

VALUE	DESCRIPTION
0 or 'No'	Shows the Wi-Fi calling error message. This is the default OS behavior.
1 or 'Yes'	Hides the Wi-Fi calling error message.

5. To show the T-Mobile specific error message in the mobile device UI, set `ShowSpecificWifiCallingError` to one of the following values:

VALUE	DESCRIPTION
0 or 'No'	Shows the Wi-Fi calling error message. This is the default OS behavior.
1 or 'Yes'	Hides the Wi-Fi calling error message.

**\*\*Note\*\*** If the mobile device is not specific to T-Mobile, OEMs should use the `GenericWifiCallingErrorMessage` setting instead.

1. To specify a custom generic Wi-Fi calling error string in the mobile device UI, set `GenericWifiCallingErrorMessage` to a string that corresponds to the error message you want to show. The string must not be longer than 127 characters.

#### Testing:

Work with your mobile operator partner to understand the Wi-Fi calling error message requirements for the operator and to test this customization on their network.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Wi-Fi calling operator name

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can customize the display name for the mobile operator when the device is using Wi-Fi calling.

**Constraints:** None

This customization supports: **per-IMSI** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="WiFiCallingOperatorName"
    Description="Use to customize the mobile operator name that's visible when the
phone
        is using Wi-Fi calling."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>

        <!-- Add the resource-only dll file and language MUI packages -->
        <Settings Path="Localization/MUI">
            <!-- Use to add your base MUI DLL file -->
            <Asset Name="BaseDll" Source="" />

            <!-- Use to specify the language MUI packages (*.dll.mui) for the languages you are supporting and
have
                localized strings for -->
            <Asset Name="LanguageDll/${langid}" Source="" />
            <Asset Name="LanguageDll/${langid}" Source="" />
            <Asset Name="LanguageDll/${langid}" Source="" />
            <!-- Add as many as you need -->
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Phone/PerSimSettings/${__IMSI}">
            <Setting Name="WiFiCallingOperatorName" Value="" />
        </Settings>
    </Variant>
</ImageCustomizations>

```

2. Specify an `Owner`.
3. Add the resource-only .dll file and the language MUI packages (\*.dll.mui) for the languages you are supporting. To do this, follow these steps:
  - a. Add the resource-only .dll that contains the custom display string by setting the `BaseDll` asset to point to the location of your base MUI DLL file. For example: C:\Path\DisplayStrings.dll.
  - b. Add the language MUI packages (\*.dll.mui) for all the languages you are supporting and have

localized strings for. To do this:

- Set the asset's `Name` to `LanguageDll/ $(langid)` where `$(langid)` corresponds to the language. For example: `LanguageDll/en-US`.
- Set the asset's `Source` to the location of the .dll.mui file for that language. For example: `C:\Path\en-us\DisplayStrings.dll.mui`.
- Repeat the previous steps for the other languages.

The following example shows the customization answer file entries for en-US, fr-CA, and es-MX languages:

```
<Asset Name="LanguageDll/en-US" Source="C:\Path\en-us\DisplayStrings.dll.mui" />
<Asset Name="LanguageDll/fr-CA" Source="C:\Path\fr-CA\DisplayStrings.dll.mui" />
<Asset Name="LanguageDll/es-MX" Source="C:\Path\es-MX\DisplayStrings.dll.mui" />
```

For more information, see [Create a resource-only .dll for localized strings](#).

#### 4. For the **per-IMSI** case:

- Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
  - Define settings for a **Variant**, which are applied if the associated target's conditions are met.
5. To customize the name of the mobile operator when the phone is using Wi-Fi calling, you can set the value for `WiFiCallingOperatorName` to:
- Use a localized MUI string – To do this, set value to the name of the resource-only .dll file and specify the string offset that corresponds to the mobile operator name. For example: `@DisplayStrings.dll,-Offset`.
  - Use a non-localized string – To do this, set value to the string that corresponds to the mobile operator name. For example: `Contoso`.

If you don't set the value for `WiFiCallingOperatorName`, the device will always display "`SIMServiceProviderName Wi-Fi`", where `SIMServiceProviderName` is a string that corresponds to the SPN for the SIM on the device. If the service provider name in the SIM is not set, only "Wi-Fi" will be displayed.

#### Testing steps:

- Flash a build containing this customization to a device that has Wi-Fi calling enabled.
- If you used a localized MUI string, verify that the localized string for the Wi-Fi calling mobile operator name is displayed on the dialer.

If you used a non-localized string, verify that this string is displayed on the dialer.

If you did not set the value, verify that "`SIMServiceProviderName Wi-Fi`" is displayed on the dialer.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Wi-Fi icon and notifications

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can configure settings related to the Wi-Fi icon.

Settings that can be configured include:

- Adjusting the percentages represented by the five bands in the status bar Wi-Fi icon.
- Modifying the minimum connection strength for displaying a Wi-Fi network to the user. The default is 15%.
- Specifying how frequently the **Wi-Fi** screen in **Settings** is updated.

**Constraints:** None

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="WiFiIconAndNotifications"
    Description="Use to configure settings related to Wi-Fi, including the percentages
    represented by the five bands in
        the status bar Wi-Fi icon and the minimum connection strength for
    displaying a Wi-Fi network to the user."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="WiFi/Config">
            <Setting Name="ScanInterval" Value="" />
            <Setting Name="SignalStrengthBar" Value="" />
            <Setting Name="SignalStrengthDelta" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set the values for the following settings:

Setting name	Description
<code>ScanInterval</code>	Specifies how often the list of available networks is updated when the user is in the <b>Wi-Fi</b> screen in <b>Settings</b> .  Set the value to the number of seconds multiplied by 1000. For example, the default is 6000 (or 6 seconds). To use a hexadecimal value, convert the decimal value to hexadecimal and add the 0x prefix.

<code>SignalStrengthBar</code>	<p>Specifies the lowest acceptable signal strength for networks to be displayed in the <b>Wi-Fi</b> screen in <b>Settings</b>.</p> <p>Set the value is to a percentage from 0 to 100. The default value is 15%. If you are using a hexadecimal value, add the 0x prefix.</p>
<code>SignalStrengthDelta</code>	<p>Specifies the difference in signal strength, as a percentage, between each bar in the Wi-Fi icon.</p> <p>Set the value to a number between 0 and 25. The default value is 15%. If you are using a hexadecimal value, add the 0x prefix.</p>

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Customizations for contacts

10/2/2018 • 2 minutes to read • [Edit Online](#)

Describes the customizations that you can configure for the contacts and contact list on the mobile device.

## In this section

TOPIC	DESCRIPTION
<a href="#">Cortana phone number</a>	Partners can configure a phone book entry for Cortana to allow users to initiate speech from a car that doesn't have support for activating speech on the device that is connected over Bluetooth.
<a href="#">Disable wait for phonebook ready signal from the modem</a>	FDN SIM contacts syncs from the SIM during device boot. By default, this component waits until the phonebook ready signal is received from the modem and then it verifies whether FDN contact management is enabled on the SIM. If needed, OEMs can disable the wait for the phonebook ready signal.
<a href="#">Hide contacts without phone numbers</a>	Partners can change the default OS behavior so that both contacts with phone numbers and contacts without phone numbers are shown in the People Hub.
<a href="#">Sort order for contacts</a>	OEMs can use this customization to set the list of contacts displayed in the People Hub to be organized by last name instead of first name, or first name instead of last name. It is also possible to change the display format of contact names to appear as "First name Last name" or "Last name First name" for markets that use more formal nomenclature.
<a href="#">Sort order for contacts override</a>	OEMs can customize the default values for people sort and display settings as documented in the <a href="#">Sort order for contacts</a> customization. However, these settings may be overridden by the defaults for the user's current locale unless the OEM sets an additional override registry key.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Cortana phone number

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can configure a phone book entry for Cortana to allow users to initiate speech from a car that doesn't have support for activating speech on the device that is connected over Bluetooth. The custom phone book entry will show up in the car's UI, but not in the Windows 10 Mobile UI.

When the user dials the custom phone number from the car, the device activates Cortana (or the device's default speech engine if Cortana is turned off or is not available), instead of making a phone call. The phone number also becomes associated with the contact name **Cortana** (or **Speech**) when the car downloads the phone book from the phone. To start the device's speech engine, the user can dial the custom contact from the car's UI or use the car's speech engine to "call Cortana" (or "call Speech" if Cortana is not enabled).

Once the phone book entry for Cortana has been configured, users cannot change the number. Only OEMs or mobile operators can change this number. Partners must set this phone number to a number that will not be used for actual phone numbers. Partners must also not use phone numbers that are used for emergencies, such as 911 in the United States.

**Constraints:** None

**Instructions:**

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="CortanaPhoneNumber"
    Description="Use to assign a phone number to associate with Cortana. Users can use
this number to start a
                                conversation with Cortana on the phone if a car doesn't have support
for activating speech on
                                the phone that is connected over Bluetooth."
    Owner=""
    OwnerType="OEM">

    <Static>
        <Settings Path="Bluetooth/BTAGService">
            <Setting Name="CortanaPhoneNumber" Value="" />
        </Settings>
    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.
3. Set the value for `CortanaPhoneNumber` to a phone number that will not be used for actual phone numbers or for those used for emergencies. The phone number should be specified for all countries/regions that support Cortana or legacy speech. If Cortana is off or not available for the market, the string **Speech** is used instead of **Cortana**.

The default Cortana phone number is currently set to a fictitious phone number, 5555559876. If you do not change the default value, the OS will only recognize this phone number.

**Testing:**

1. Flash an image that contains this customization to a phone.

2. Connect your phone to your car over Bluetooth.
3. If Cortana or the default phone speech engine is supported, start the phone's speech engine by dialing the custom contact from the car's UI. Alternatively, you can use the car's speech engine and say "call Cortana" (or "call Speech) if Cortana is not enabled).

Verify that the phone activates either Cortana or the default speech instead of making a phone call. If Cortana is available, start a conversation.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Disable wait for phonebook ready signal from the modem

10/2/2018 • 2 minutes to read • [Edit Online](#)

FDN SIM contacts syncs from the SIM during device boot. By default, this component waits until the phonebook ready signal is received from the modem and then it verifies whether FDN contact management is enabled on the SIM. If needed, OEMs can disable the wait for the phonebook ready signal.

**Constraints:** ImageTimeOnly

**Instructions:**

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="CheckFDNStateAfterPhonebookReady"
    Description="Use to disable the wait for the phonebook ready signal from the
modem."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="People/SIMContactManagement">
            <Setting Name="CheckFDNStateAfterPhonebookReady" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set the value of `CheckFDNStateAfterPhonebookReady` to one of the following values:

VALUE	DESCRIPTION
0 or 'False'	Disables the wait until the phonebook ready signal from the modem is received.
1 or 'True'	Waits for the phonebook ready signal from the modem before verifying whether FDN contact management is enabled on the SIM.

**Testing steps:**

1. Flash the build containing this customization to a device with a SIM that has FDN enabled.
2. Go through the setup process and then enter the PIN to unlock the SIM when prompted and wait for the Start screen to appear.
3. Go to the **People** Hub and verify that FDN contacts are visible.

4. Go to the **Settings > phone > SIM** settings screen and verify that FDN is shown as On.
5. Additionally, you can test SIMs from two operators and verify that:
  - Both SIM cards show FDN contacts correctly.
  - Enabling and disabling FDN works.
  - Operator voice calls are allowed or blocked correctly based on the FDN status and FDN contacts list.
  - Adding and deleting contacts in the FDN phonebook works.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Hide contacts without phone numbers

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can change the default OS behavior so that both contacts with phone numbers and contacts without phone numbers are shown in the People Hub.

By default, contacts that do not have phone numbers are hidden in the People Hub.

**Constraints:** ImageTimeOnly

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="HideContactsWithoutPhoneNumbers"
    Description="Use to show or hide in the People Hub the contacts without phone
numbers."
    Owner=""
    OwnerType="OEM">
    <Static>
        <Settings Path="People/ContactsFilteringSettings">
            <Setting Name="HideContactsWithoutPhoneNumbers" Value="" />
        </Settings>
    </Static>
</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set `HideContactsWithoutPhoneNumbers` to one of the following values:

VALUE	DESCRIPTION
1 or 'True'	In the People Hub, this hides contacts without phone numbers. This is the default OS behavior.
0 or 'False'	In the People Hub, this shows contacts with phone numbers and contacts without phone numbers.

## Testing steps:

1. Flash a build containing this customization to a device.
2. Set up the device and then add a few contacts. Make sure to include contacts with phone numbers and contacts without phone numbers.
3. Open the People Hub.
  - If you set `HideContactsWithoutPhoneNumbers` to 0 or 'False', verify that under the **Contacts** heading the filter shows **showing all** at the top of the contacts list. Confirm that all contacts, with and without phone numbers, are showing.

- If you set `HideContactsWithoutPhoneNumbers` to 1 or 'True' (or did not set this setting), verify that under the **Contacts** heading the filter shows **showing contacts with phone numbers** at the top of the contacts list. Confirm that only contacts with phone numbers are showing.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Sort order for contacts

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can use this customization to set the list of contacts displayed in the People Hub to be organized by last name instead of first name or first name instead of last name. It is also possible to change the display format of contact names to appear as "First name Last name" or "Last name First name" for markets that use more formal nomenclature.

**Constraints:** ImageTimeOnly

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="ContactSortSettings"
    Description="Use to set the sorting and display setting of the user's contacts"
    Owner=""
    OwnerType="OEM">

<Static>
    <Settings Path="People/SortAndDisplaySettings">
        <Setting Name="SortBy" Value="" />
        <Setting Name="DisplayBy" Value="" />
    </Settings>
</Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set the `SortBy` Value to one of the following values:

VALUE	DESCRIPTION
[Sort1] or FirstLast	Sorts the contacts in the People Hub by their first name.
[Sort2] or LastFirst	Sorts the contacts in the People Hub by their last name.

4. Set the `DisplayBy` Value to one of the following values:

VALUE	DESCRIPTION
[Sort1] or FirstLast	Displays the contacts in the People Hub in the format: "First name Last name"
[Sort2] or LastFirst	Displays the contacts in the People Hub in the format: "Last name First name".

**Testing steps:**

1. Flash a build containing this customization to a device.
2. Go to the **People** settings screen. Verify that the **Sort list by** option is set to **Last name**, and that the **Display names by** option is set to **Last, First**.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Sort order for contacts override

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can customize the default values for people sort and display settings as documented in the [Sort order for contacts](#) customization. However, these settings may be overridden by the defaults for the user's current locale unless the OEM sets an additional override registry key.

**Constraints:** ImageTimeOnly

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="OEMOverridesSortDisplay"
    Description="Use to prevent OEM values for people sort and display settings from
    being overridden by user's current locale."
    Owner=""
    OwnerType="OEM">
<Static>
    <Settings Path="People/SortAndDisplaySettings">
        <Setting Name="OEMOverridesSortDisplay" Value="" />
    </Settings>
</Static>

</ImageCustomizations>
```

2. Specify an `Owner`.
3. Set the `OEMOverridesSortDisplay` value to 1 or 0x1 to prevent the OEM values for people sort and display settings from being overridden.

## Testing steps:

1. Flash a build containing this customization to a device.
2. Go to the **People** settings screen.
3. Verify that the **Sort list by** and the **Display names by** option is set to the values you specified in the [Sort order for contacts](#) customization.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Customizations for desktop experiences

10/2/2018 • 2 minutes to read • [Edit Online](#)

Describes the customizations that you can configure for the desktop when the mobile device is connected.

## In this section

TOPIC	DESCRIPTION
<a href="#">Control Panel device icon</a>	OEMs can change the default icon associated with the phone on a connected computer.
<a href="#">Phone image in the phone app</a>	OEMs can replace the default images of the phone that appears in the phone app. These images are included in the OEMImage.cab that is provided in this customization sample. The OEM can replace these images with custom ones that more accurately depict their phone. When the OEM provides a new image, this image will be used and will replace the OEMAvatar.cab file that is used by default.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Control Panel device icon

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can change the default icon associated with the phone on a connected computer.

When the user connects a phone to a Windows computer, the phone name and icon shows up in the **Devices and Printers** list. OEMs can change the default icon associated with the phone.

## Limitations and restrictions:

Create an icon to represent the phone that meets the following specifications:

- **Filename:** Device.ico

- **Dimensions, bit level and transparency:**

256x256: 32bit + Alpha

48x48: 32bit + Alpha

48x48: 8bit 256

48x48: 8bit 16

32x32: 32bit + Alpha

32x32: 8bit 256

32x32: 4bit 16

24x24: 32bit + Alpha

24x24: 8bit 256

24x24: 4bit 16

16x16: 32bit + Alpha

16x16: 8bit 256

16x16: 4bit 16

- Match the orientation and general creative style of the sample image. To avoid issues associated with the localization of the screen image text, the phone image must depict a phone that is turned off.

## Constraints:

### Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="ControlPanelDeviceIcon"
    Description="Use to change the icon associated with the phone when connecting to a
Windows computer."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="MediaTransferProtocol/DeviceAssets">
            <!-- Use to add the icon to represent the phone when connected to a Windows computer. -->
            <Asset Name="DeviceIcon" Source="C:\Path\Device.ico" />

            <!-- Use to specify the file name of the device icon to use -->
            <Setting Name="Icon" Value="Device.ico" />
        </Settings>

    </Static>

</ImageCustomizations>

```

2. Specify an `Owner`.
3. Add the icon you want to use to represent the phone when connected to a Windows computer. To do this:
  - a. Set the asset `Name` to **DeviceIcon**.
  - b. Specify the file name and location of the asset on your workstation by setting the `Source`.
4. Set the `Icon` value to the name of your custom icon. For example, *Device.ico*.

**Testing steps:**

1. Flash a build containing this customization to a phone.
2. On the Windows computer, open **Device Manager** and remove the driver associated with the phone.
3. Connect the phone to the computer using a USB cable.
4. On the computer, navigate to the **Devices and Printers** screen. Verify that the phone icon that you included in the build is visible.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Phone image in the phone app

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can replace the default images of the phone that appears in the phone app. These images are included in the OEMImage.cab that is provided in this customization sample. The OEM can replace these images with custom ones that more accurately depict their phone. When the OEM provides a new image, this image will be used and will replace the OEMAvatar.cab file that is used by default.

## Limitations and restrictions:

The custom image files to represent the phone must meet the following specifications:

- PNG24
- RGB
- Bicubic
- 96 DPI
- Alpha background transparency
- Dimensions - Height (width varies by device):

800 px

400 px

200 px

150 px

120 px

80 px

- The image must be the exact height and width of the device with no padding.
- Match the orientation and general creative style of the sample image. To avoid issues associated with the localization of the screen image text, the phone image must depict a phone that is turned off.

**Constraints:** None

## Instructions:

1. Complete the following steps to create a cab file containing six custom .png image files to represent the phone.
  - a. Create the six custom phone images and place them in a folder named OEMImage.
  - b. Create a new OEMImage.cab file that contains your custom images using Makecab.exe, which is a utility included with Windows. To do this:
    - a. Create a directive file for Makecab.exe with the filename OEMImage.ddf.

The .ddf file must have the following contents:

```

;*** OEMImage.ddf example
;
.OPTION EXPLICIT      ; Generate errors
.Set CabinetNameTemplate=OEMImage.cab
.set DiskDirectoryTemplate=CDROM ; All cabinets go in a single directory
.Set UniqueFiles="OFF"
.Set Cabinet=on
.Set DiskDirectory1=OEMImage.CAB
DeviceImage_80.png
DeviceImage_120.png
DeviceImage_150.png
DeviceImage_200.png
DeviceImage_400.png
DeviceImage_800.png

```

- b. Place the OEMImage.ddf file in the OEMImage folder along with the six .png image files. At a command-line prompt in the OEMImage folder, run the following command:

```
Makecab.exe /F OEMImage.ddf
```

This produces a .cab file called OEMImage.cab.

2. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="WindowsPhoneAppImage"
    Description="Use to replace the default images of the phone that appears in the
Windows Phone app."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="MediaTransferProtocol/DeviceAssets">
            <!-- Use to add the .cab containing the PNG image files that depict the phone at various sizes
and the OEMImage.ddf -->
            <Asset Name="DeviceImageCab" Source="C:\Path\OEMImage.cab" />

            <!-- Use to specify which device image .cab should be used to display images of the phone in
the Windows Phone app -->
            <Setting Name="Avatar" Value="OEMImage.cab" />
        </Settings>

    </Static>

</ImageCustomizations>

```

3. Specify an **Owner**.

4. Add the OEMImage.cab file to the customization package. To do this:

- Set the asset **Name** to **DeviceImageCab**.
- Specify the location of the OEMImage.cab on your workstation by changing C:\Path\ in the **Source** attribute to match the location of OEMImage.cab.
- Specify which device image .cat to use when displaying images of the phone in the Windows Phone app by setting the **Avatar** value to OEMImage.cab.

#### **Testing steps:**

1. Flash a build containing this customization to a phone.
2. On the computer, install the Windows 10 Mobile app.
3. Connect the phone to the computer using a USB cable.
4. On the computer, open the mobile app.
5. Verify that the phone image that you included in the build is visible in the mobile app.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Customizations for display and lock screen

10/2/2018 • 2 minutes to read • [Edit Online](#)

Describes the customizations for the display and lock screen on the device.

## In this section

TOPIC	DESCRIPTION
<a href="#">Additional lock screen backgrounds</a>	OEMs can add new lock screen background images for the lock screen and also set the default lock screen background.
<a href="#">Brightness tuning</a>	When the <b>Brightness</b> screen in <b>Settings</b> is not set to automatically adjust, this customization enables the user to select low, medium, and high intensities for the screen brightness.
<a href="#">Default theme and accent color for Kid's Corner</a>	Partners can set the default theme, including the background color (light or dark) and the accent color for Kid's Corner.
<a href="#">Enable dark mode</a>	OEMs can choose to specify whether the dark mode is applied.
<a href="#">Hide the auto brightness setting</a>	OEMs can hide the automatic brightness setting for phones that do not have an ambient light sensor.
<a href="#">Lock screen notifications</a>	OEMs can preload apps that support lock screen notifications.
<a href="#">Lock screen timeout for AMOLED and OLED displays</a>	OEMs can remove the 15 minutes, 30 minutes, and Never options from the Screen times out after dropdown in the Lock screen settings screen.
<a href="#">Warning about light theme for AMOLED and OLED screens</a>	OEMs can choose to display a warning about battery life if the user selects the light theme on phones with AMOLED or OLED displays. This customization is not relevant for other screen types, as there is not a significant power difference between the themes with the light and dark background.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Additional lock screen backgrounds

10/2/2018 • 3 minutes to read • [Edit Online](#)

OEMs can add new lock screen background images for the lock screen and also set the default lock screen background.

## Limitations and restrictions:

- The lock screen backgrounds included by Microsoft shall not be removed, moved, or altered.
- The user can modify the default lock screen background by selecting a new photo from the Photos application or by changing the lock screen background settings.
- Lock screen background images must be in .JPG, .JPEG, or .PNG format.

**Constraints:** FirstVariationOnly

## Instructions:

### Adding more lock screen backgrounds

OEMs can provide a set of images to complement the Backgrounds album. The set of images will be available to end-users to select as a lock screen background when the user selects the **Sample images** background provider in the **Lock screen** settings screen and launches the photo picker. The images will be appended to the end of the backgrounds album. Although there is no limit to the number of additional lock screen backgrounds, we recommend that partners add no more than 26 backgrounds.

To add a set of lock screen backgrounds:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="AdditionalLockScreenBackgrounds"
    Description="Use to add additional lock screen backgrounds and set the default lock
screen background."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="LockScreen">
            <Asset Name="Wallpapers" Source="" />
            <Asset Name="Wallpapers" Source="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

If you are setting the default lock screen background in addition to adding a set of lock screen backgrounds, see *Setting the default system lock screen background* after this section.

2. Specify an **Owner**.
3. Add additional lock screen backgrounds by adding more **Asset** elements.
  - Set the asset's **Name** to **Wallpapers**.

- Set **Source** to the full path to the lock screen background source file on your development machine.

For example: C:\Program Files (x86)\Windows

Kits\10\OEMCustomizationAssets\AdditionalLockScreenBackgrounds\Image1.jpg.

## Setting the default system lock screen background

OEMs can set an image to be the default system lock screen background and this supersedes the default device system lock screen background. This image will be shown during the following scenarios:

- During first boot, unless device restore overrides the image with a backed up image.
- When **Sample images** is selected as the **Background** in the **Lock screen** settings screen and no custom image is assigned.
- During any error condition when the lock screen API fails to set the image and the system rolls back to the default system image.

The default lock screen image can be one of the images supplied as the default lock screen background set to be shown on the photo picker. We recommend that the default lock screen background image also be part of the set.

### To configure the default lock screen background:

- Create a customization answer file using the contents shown in the following code sample or use the sample AdditionalLockScreenBackgrounds.xml file.

```
<Settings Path="LockScreen">
  <Setting Name="DefaultWallpaper" Value="" />
</Settings>
```

- Specify an **Owner**.
- Set the **DefaultWallpaper** value to the file name of the image that you want to set as the default lock screen.  
For example, *Image1.jpg*.
- Use the customization answer file to create your custom OS image.

### Testing:

- Flash an OS image that contains this customization to a phone.
- If you added more lock screen backgrounds:
  - Go to **Lock screen** settings screen.
  - Make sure the **Background** setting is set to **Sample images**.
  - Select **Browse** and verify that your additional lock screen backgrounds have been appended at the end.
- If you set the default lock screen background:
  - Turn off the phone then turn it on again.
  - Verify that the lock screen background matches the image that you set as the default lock screen.

## Related topics

[Prepare for Windows mobile development](#)

Customization answer file overview](<https://docs.microsoft.com/en-us/windows-hardware/customize/mobile/mcsf/customization-answer-file>)

# Brightness tuning

10/2/2018 • 2 minutes to read • [Edit Online](#)

When the **Brightness** screen in **Settings** is not set to automatically adjust, this customization enables the user to select low, medium, and high intensities for the screen brightness. By default these values are 33%, 66%, and 100% of maximum brightness respectively. OEMS can tune these values so that low intensity is as close to 35 LUX as possible and medium is as close to 200 LUX as possible. High intensity should be the screen's maximum supported brightness.

## Limitations and restrictions:

- When the **Brightness** setting is set to automatically adjust, partners must ensure that the brightness table (sometimes called curve) provided for automatic brightness setting contains at least one entry that is less than 33% of the maximum brightness. If this requirement is not met, the behavior of the device when switching to low brightness settings is undefined.

**Constraints:** ImageTimeOnly

## Instructions:

- Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="BrightnessTuning"
    Description="Use to tune the phone's low, medium, and high brightness settings when
    the brightness is not set to automatically adjust."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Brightness">
            <Setting Name="LowBrightnessValue" Value="" />
            <Setting Name="MediumBrightnessValue" Value="" />
            <Setting Name="HighBrightnessValue" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

- Specify an **Owner**.

- Set the brightness percentages. Note that you will have to convert the decimal percentage to hexadecimal and then prepend 0x to the hexadecimal value. For example, if the value is 200 (decimal), the final hexadecimal value must be set to 0xC8.

SETTING	DESCRIPTION
LowBrightnessValue	Use to set the low brightness percentage. Low intensity should be as close to 35 LUX as possible.

SETTING	DESCRIPTION
MediumBrightnessValue	Use to set the medium brightness percentage. Medium intensity should be as close to 200 LUX as possible.
HighBrightnessValue	Use to set the high brightness percentage. High intensity should remain 0x64 (100 in decimal) to use the screen's maximum supported brightness.

### Testing:

1. Flash the build containing this customization to a phone.
2. Go to the **theme** screen in **Settings**, and select the **light** background.
3. Go to the **Brightness** screen in **Settings**, and turn off the **Automatically adjust** option.
4. Select **low** level, then use a light meter placed over the bottom half of the screen (which is all white) to test the output in LUX. For best results the light sensor should be surrounded by a shroud to block out incidental light. Low should be approximately equivalent to 35 LUX.
5. Repeat the test by selecting the **medium** level. This should be approximately equivalent to 200 LUX.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Kid's Corner default theme and accent color

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can set the default theme, including the background color (light or dark) and the accent color for Kid's Corner.

**Constraints:** FirstVariationOnly

**Instructions:**

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="KidsCornerThemeAndAccent"
    Description="Use to set the default theme and accent color for Kid's Corner."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="KidsCorner">
            <Setting Name="DefaultThemeBackground" Value="" />
            <Setting Name="DefaultThemeAccentColor" Value="" />
        </Settings>

    </Static>
</ImageCustomizations>
```

2. Specify an `Owner`.
3. Set `DefaultThemeBackground` to specify a theme. The possible values are:
  - **0** for light
  - **1** for dark
4. Set `DefaultThemeAccentColor` to specify an accent color ID. You can use either the corresponding decimal value or the corresponding hexadecimal value (with the 0x prefix) as shown in the following table.

ACCENT COLOR ID (IN HEXADECIMAL)	ACCENT COLOR ID (IN DECIMAL)	COLOR NAME
0x0	0	Lime
0x1	1	Green
0x2	2	Emerald
0x3	3	Teal
0x4	4	Cyan

ACCENT COLOR ID (IN HEXADECIMAL)	ACCENT COLOR ID (IN DECIMAL)	COLOR NAME
0x5	5	Cobalt
0x6	6	Indigo
0x7	7	Violet
0x8	8	Pink
0x9	9	Magenta
0xA	10	Crimson
0xB	11	Red
0xC	12	Orange
0xD	13	Amber
0xE	14	Yellow
0xF	15	Brown
0x10	16	Olive
0x11	17	Steel
0x12	18	Mauve
0x13	19	Taupe
0x65	101	CustomAccentColor1
0x66	102	CustomAccentColor2
0x67	103	CustomAccentColor3
0x68	104	CustomAccentColor4

**Testing:**

1. Flash the build containing this customization to a device.
2. Launch **Kid's Corner**.
3. Verify that the Kid's Corner theme and accent color match the defaults you set in the registry.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Start + theme settings: Enabling dark mode

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can choose to specify whether the dark mode is applied.

**Constraints:** FirstVariationOnly

**Instructions:**

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="DefaultBackgroundColor"
    Description="Specifies whether the dark mode is applied."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Theme">
            <Setting Name="DefaultBackgroundColor" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set `DefaultBackgroundColor` to specify a mode. The possible values are:

VALUE	DESCRIPTION
0 or 'Light'	Enables light mode.
1 or 'Dark'	Enables dark mode. This is the default OS setting.

**Testing:**

1. Flash an image containing this customization to a phone.
2. Verify that the phone is using the correct mode.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Hide the auto brightness setting

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can hide the automatic brightness setting for phones that do not have an ambient light sensor.

**Constraints:** ImageTimeOnly

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="HideAutoBrightness"
    Description="Use to hide the auto brightness setting for phones without an ambient
light sensor."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Brightness">
            <Setting Name="HideAutoBrightness" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set `Value` to one of the following:

VALUE	DESCRIPTION
0 or 'Show'	Use to show the <b>Automatically adjust</b> setting in the <b>Settings &gt; brightness</b> screen.
1 or 'Hide'	Use to hide the <b>Automatically adjust</b> setting in the <b>Settings &gt; brightness</b> screen.

## Testing:

1. Flash the build that contains this customization to a phone.
2. In **Settings**, go to the **Brightness** screen.
3. Verify that the **Automatically adjust** toggle is hidden or visible depending on the `value` that you set for `HideAutoBrightness`.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Lock screen notifications

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can preload apps that support lock screen notifications. Notifications alert the user to new content or updates in that app. They can take the form of a number (to indicate the number of changes) or a text preview.

OEMs can also preset one notification in the fifth slot on the lock screen. This notification displays an icon for a single app plus the number of notifications for that app.

## Limitations and restrictions:

- The default count notification mappings must not be changed.
- The default text notification mapping must not be changed.
- Notification icons provided by the app and displayed on the lock screen must be monochromatic, with a white foreground color and a transparent background color.
- After first boot, apps can only be assigned as a lock screen notification by the user; apps must not automatically use a notification slot.
- Users can delete apps that support notifications, and they can also select alternate notification mappings.

**Constraints:** FirstVariationOnly

## Instructions:

For more information about building an app that supports notifications on the lock screen, see the Windows developer documentation.

## To preload and specify the app that supports lock screen notifications:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="LockScreenNotifications"
    Description="Use to preload an app that supports lock screen notifications and set
the app to use the 5th slot on the lock screen."
    Owner=""
    OwnerType="OEM">

    <Static>
        <Applications>
            <Application Source=""
                License=""
                ProvXML="" />
        </Applications>

        <Settings Path="LockScreen">
            <Setting Name="LockNotificationAppID" Value="" />
            <Setting Name="LockNotificationAppTile" Value="default" />
        </Settings>
    </Static>
</ImageCustomizations>
```

2. Specify an `Owner`.

3. Preload the app that supports lock screen notifications by specifying these settings:

- Set `Source` to the path and name of the app's .xap file.
- Set `License` to the path and name of the app's license file.
- Set `ProvXML` to the path and name of the app's PROVXML file.

4. Set the `LockNotificationAppID` setting `Value` to correspond to your app's app ID or GUID.

5. Replace the `LockNotificationAppTile` setting `Value` to match the **TokenID** value that was in your app manifest file. If you do not have a **TokenID** specified, set the `Value` to *default*.

#### Testing:

1. Flash an image that contains this customization to a device.
2. Go to **Settings > Lock screen**.
3. Scroll down and verify that your lock screen notification app now shows as the default app in fifth slot under **Choose apps to show quick status**.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Screen timeout for AMOLED and OLED displays

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can remove the **15 minutes**, **30 minutes**, and **Never** options from the **Screen times out after** dropdown in the **Lock screen** settings screen.

This is recommended for phones with AMOLED and OLED screens to prevent screen damage.

**Constraints:** FirstVariationOnly

**Instructions:**

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="ScreenTimeout"
    Description="Use to remove the 15 minutes, 30 minutes, and Never options from the
    screen time-out option in the Lock screen settings screen."
    Owner=""
    OwnerType="OEM">

    <Static>
        <Settings Path="LockScreen">
            <!-- Set the value to 1 or 0x1 to remove the 15 minutes, 30 minutes, and Never options from the
            lock screen settings screen -->
            <Setting Name="ScreenTimeOut" Value="" />
        </Settings>
    </Static>
</ImageCustomizations>
```

2. Specify an `Owner`.
3. To hide or remove the **15 minutes**, **30 minutes**, and **Never** options from the **Screen times out after** dropdown in the **Lock screen** settings screen, set `ScreenTimeOut` to 1 or 0x1.

**Testing:**

1. Flash the build containing this customization to a phone.
2. Go to the **lock screen** screen in **Settings**.
3. Tap the **Screen times out after** setting and verify that the **15 minutes**, **30 minutes**, and **never** options are no longer included in the list.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Warning about light theme for AMOLED and OLED screens

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can choose to display a warning about battery life if the user selects the light theme on phones with AMOLED or OLED displays. This customization is not relevant for other screen types, as there is not a significant power difference between the themes with the light and dark background.

**Constraints:** FirstVariationOnly

**Instructions:**

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="OLEDWarning"
    Description="Use to display the battery life warning on phones with AMOLED or OLED
    displays."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Theme">
            <Setting Name="OLEDWarning" Value="1" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.
3. Do not modify the `Value`.

**Testing:**

1. Flash an image containing this customization to a phone.
2. Go to the **Colors** settings screen.
3. Set the **Choose your mode** option to **Light**.
4. Verify that the warning text appears above the mode option.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Customizations for email

10/2/2018 • 2 minutes to read • [Edit Online](#)

Describes the customizations related to email.

## In this section

TOPIC	DESCRIPTION
<a href="#">Light or dark theme in email</a>	Partners can specify that the entire email application always has a light background.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Light or dark mode in email

10/2/2018 • 2 minutes to read • [Edit Online](#)

The email application consists of several screens, including the List View, Settings, Automatic Replies, Search, Inbox Linking, Read, and Compose. By default, the email background for all pages except Read and Compose match the system theme. The Read and Compose pages always have a light background. Partners can specify that the entire email application always has a light background, but users do not have access to this setting. However, if the user turns on high-contrast mode, it overrides all other settings, and all screens use the high contrast settings.

**Constraints:** None

**Instructions:**

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="DefaultEmailBackgroundTheme"
    Description="Use to configure the entire email app to always have a light
background."
    Owner=""
    OwnerType="OEM">

    <Static>
        <Settings Path="Email">
            <Setting Name="DefaultBackgroundThemeSetting" Value="" />
        </Settings>
    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set `DefaultBackgroundThemeSetting` to one of the following values:

VALUE	DESCRIPTION
1 or Light Background	Configures the entire email application to always have a light background.  Users cannot override this setting, but if high contrast is enabled this setting will be changed.
0 or System Default	Keeps the default background theme for the email app.

**Testing steps:**

1. Flash the build containing this customization to a phone that has data connectivity.
2. Go to the **theme** screen in **Settings**. Set the background theme to dark.
3. Go to the **email+accounts** screen in **Settings**. Configure an email account.
4. Go to the configured email account's inbox.

5. Verify that the email background is light.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Customizations for keyboard

10/2/2018 • 2 minutes to read • [Edit Online](#)

Describes the customizations that you can configure for the keyboard on the mobile device.

## In this section

TOPIC	DESCRIPTION
<a href="#">Disable text correction and suggestions</a>	For markets that do not use any of the available input languages, partners pick an alternative available input language as the default, but disable text prediction, auto-correction, and the spelling checker by default, using this customization.
<a href="#">Hardware keyboard character repeats hold time and delay</a>	For devices that have a hardware keyboard, partners can optionally set the character repeat hold time and delay.
<a href="#">On-screen keyboard delay</a>	When an external keyboard (e.g. Bluetooth keyboard or barcode scanner which connects as an HID keyboard) is used with a device, the on-screen keyboard is hidden. When the screen is touched, there is a hard-coded delay of 60s before the on-screen keyboard reappears. With this customization, an OEM can define the delay value before the on-screen keyboard reappears.
<a href="#">Pre-enabled keyboard</a>	OEMs can use this customization to pre-enable additional device keyboards.
<a href="#">Text correction and suggestions</a>	Partners must enable text correction and text suggestions for at least one input language, and can optionally include more.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Disable text correction and suggestions

10/2/2018 • 2 minutes to read • [Edit Online](#)

For markets that do not use any of the available input languages, partners pick an alternative available input language as the default, but disable text prediction, auto-correction, and the spelling checker by default, using this customization.

Partners must not remove any keyboards, disable default keyboards for an input language, nor modify any keyboard layouts in any way. However, for markets that do not use any of the available input languages, partners pick an alternative available input language as the default, but disable text prediction, auto-correction, and the spelling checker. This prevents the user's words from being automatically changed to similar words in the alternate language as they type.

Users can turn text prediction, auto-correction, and the spelling checker back on by going to the **keyboard** screen in **Settings**, tapping their desired keyboard language, and selecting **Suggest text**.

## Limitations and restrictions:

- Partners can disable text correction and suggestions for only one language. This functionality cannot be disabled for Japanese, Chinese, or Korean.

**Constraints:** None

## Instructions:

- Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="DisableTextCorrection"
    Description="Use to disable text prediction, auto-correction, and spelling checker
for an alternative input language."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="TextInput/Intelligence">
            <Setting Name="DisablePredictions" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

- Specify an `Owner`.
- Set the `value` to the locale or alternative input language that must have the text intelligence features disabled. For example, to disable text correction and suggestions for English (UK), set `value` to `en-gb`.

## Testing Steps:

- Flash the build containing this customization to a device.
- Go to the **keyboard** screen in **Settings**.
- Choose the keyboard language where you turned off text correction and suggestions to open the keyboard

settings.

4. Verify that the checkboxes for **Suggest text**, **Highlight misspelled words**, **Correct misspelled words**, and **Insert a space after selection a suggestion** are unchecked.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Hardware keyboard character repeats hold time and delay

10/2/2018 • 2 minutes to read • [Edit Online](#)

For devices that have a hardware keyboard, partners can optionally set the character repeat hold time and delay.

The optional keyboard customizations are:

- The amount of time, in milliseconds, the user must hold down a key before the keyboard character repeats.
- The amount of time, in milliseconds, to use as the delay between each keyboard character repeats.

**Constraints:** ImageTimeOnly

**Instructions:**

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="HardwareKeyboardSettings"
    Description="Use to configure the settings for the hardware keyboard character
repeats hold time and delay"
    Owner=""
    OwnerType="OEM">

<Static>
    <Settings Path="TextInput/HardwareKeyboard">
        <Setting Name="AutoRepeatInitialDelay" Value="" />
        <Setting Name="KeyRepeatRate" Value="" />
    </Settings>
</Static>

</ImageCustomizations>
```

2. Specify an `Owner`.
3. To specify the amount of time that the user must hold down a key before the keyboard character repeats, set the value for the `AutoRepeatInitialDelay` setting. The value is in milliseconds.
4. To specify the amount of time to use as the delay between each keyboard character repeats, set the value for the `KeyRepeatRate` setting. The value is in milliseconds.

**Testing Steps:**

1. Flash the build containing this customization to a phone with a hardware keyboard.
2. Set up the phone. Press and hold down the same keyboard key. Verify that the amount of time needed to hold down a key before the keyboard character repeats corresponds to the value that you set. Also verify that the delay between each character repeat is equivalent to the value that you set.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# On-screen keyboard delay

10/2/2018 • 2 minutes to read • [Edit Online](#)

When an external keyboard (e.g. Bluetooth keyboard or barcode scanner which connects as an HID keyboard) is used with a device, the on-screen keyboard is hidden. When the screen is touched, there is a hard-coded delay period of 60 seconds before the on-screen keyboard reappears. Typically, the user selects an input field on the screen and then uses the external keyboard/barcode scanner to enter the data.

This customization enables an OEM to change that delay period. The delay value is read on boot-up and cannot be changed during runtime. The optional customization specifies the delay in seconds with a default value of 60. If the value is set to 0, then the on-screen keyboard is not hidden.

**Constraints:** ImageTimeOnly

## Instructions

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="HWKeyboardActivityInterval"
    Description="Use to configure the setting for the delay in showing the on-screen keyboard when an
    external keyboard is connected"
    Owner=""
    OwnerType="OEM">
    <Static>
        <Settings>
            <Setting Name="HWKeyboardActivityInterval" Value="" />
            <RegistrySource Type="REG_DWORD"
                Path="HKEY_LOCAL_MACHINE\Software\Microsoft\Input\HWKeyboardActivityInterval" />
        </Settings>
    </Static>
</ImageCustomizations>
```

2. Specify an Owner.
3. Set the value to the required delay between when a user touches the screen of the device, and when the on-screen keyboard appears, in seconds.

## Testing steps

1. Flash the build containing this customization to a device
2. Connect an external Bluetooth keyboard to the device
3. Select the input field on the screen via the on-screen keyboard
  - a. Type **abc** via the on-screen keyboard.
  - b. Type **dce** via the external keyboard. The on-screen keyboard shrinks from the screen.
  - c. Touch the screen again and confirm the on-screen keyboard remains hidden until the specified delay expires.

## Related topics

[Prepare for Windows mobile development](#)

## Customization answer file overview

# Pre-enabled keyboard

10/2/2018 • 5 minutes to read • [Edit Online](#)

OEMs can use this customization to pre-enable additional device keyboards.

During device bring-up, OEMs must set the boot locale, or default locale, for the device. During first boot, the OS reads the locale setting and automatically enables a default keyboard based on the locale to keyboard mapping table in [Set languages and locales](#).

The mapping works for almost all regions and additional customizations are not needed unless specified in the pre-enabled keyboard column in [Set languages and locales](#). If an OEM chooses to pre-enable more keyboards for a particular market, they can do so by specifying this setting. Pre-enabled keyboards will automatically be enabled during boot. Microsoft recommends that partners limit the number of pre-enabled keyboards to those languages that correspond to the languages spoken within the market.

**Constraints:** None

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="PreEnabledKeyboard"
    Description="Use to pre-enable more keyboards for a particular market and specify
the keyboards automatically enabled during boot."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="TextInput">

            <!-- Replace $(InputMethodId) with the keyboard to pre-enable. Format is "LocaleCode.LocaleValue"
                For example, set Setting Name to "PreEnabledKeyboard/ko-KR.4" to pre-enable Korean 12-key
                Sky.
                Value is always set to 1 to enable the keyboard. -->
            <Setting Name="PreEnabledKeyboard/${InputMethodId}" Value="1" />

            <!-- Add additional keyboards
            <Setting Name="PreEnabledKeyboard/${InputMethodId}" Value="1" />
            <Setting Name="PreEnabledKeyboard/${InputMethodId}" Value="1" />
            -->

        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Replace `$(InputMethodId)` with the keyboard you want to pre-enable. The format must be: **Locale code.Locale value**. See the following table for more information on the locale codes and values that you can use. The setting `Value` must be set to 1 to enable the keyboard.

For example, to pre-enable US English and Korean 12-key Sky, add the following key/value pairs:

```

<Settings Path="TextInput">
    <Setting Name="PreEnabledKeyboard/en-US.1" Value="1" />
    <Setting Name="PreEnabledKeyboard/ko-KR.4" Value="1" />
</Settings>

```

The following table shows the values that you can use for the `$(InputMethodId)` part of the setting name. Replace `$(InputMethod)` with this format: **Locale code.Locale value**

### Note

The keyboards for some locales require additional language model files. To pre-enable the keyboards for the following locales, OEMs must include corresponding locales in the *Keyboard* section described in [Text correction and suggestions](#): am-ET, bn-IN, gu-IN, hi-IN, ja-JP, kn-IN, ko-KR, ml-IN, mr-IN, my-MM, or-IN, pa-IN, si-LK, ta-IN, te-IN, zh-TW, zh-CN, and zh-HK.

NAME	LOCALE CODE	KEYBOARD LAYOUT VALUE
Afrikaans (South Africa)	af-ZA	1
Albanian	sq-AL	1
Amharic	am-ET	1
Arabic	ar-SA	1
Armenian	hy-AM	1
Assamese - INSCRIPT	as-IN	1
Azerbaijani (Cyrillic)	az-Cyrl-AZ	1
Azerbaijani (Latin)	az-Latn-AZ	1
Bangla (Bangladesh) - 49 key	bn-BD	1
Bangla (India) - INSCRIPT	bn-IN	1
Bangla (India) - Phonetic	bn-IN	2
Bashkir	ba-RU	1
Basque	eu-ES	1
Belarusian	be-BY	1

<b>NAME</b>	<b>LOCALE CODE</b>	<b>KEYBOARD LAYOUT VALUE</b>
Bosnian (Cyrillic)	bs-Cyrl-BA	1
Bosnian (Latin)	bs-Latn-BA	1
Bulgarian	bg-BG	1
Catalan	ca-ES	1
Central Kurdish	ku-Arab-IQ	1
Cherokee	chr-Cher-US	1
Chinese Simplified QWERTY	zh-CN	1
Chinese Simplified - 12-key	zh-CN	2
Chinese Simplified - Handwriting	zh-CN	3
Chinese Simplified - Stroke	zh-CN	4
Chinese Traditional (Hong Kong SAR) - Cangjie	zh-HK	1
Chinese Traditional (Hong Kong SAR) - Quick	zh-HK	2
Chinese Traditional (Hong Kong SAR) - Stroke	zh-HK	3
Chinese Traditional (Taiwan) - BoPoMoFo	zh-TW	1
Chinese Traditional (Taiwan) - Handwriting	zh-TW	2
Croatian	hr-HR	1
Czech	cs-CZ	1

NAME	LOCALE CODE	KEYBOARD LAYOUT VALUE
Danish	da-DK	1
Divehi	dv-MV	1
Dutch (Belgium)	nl-BE	1
Dutch (Netherlands)	nl-NL	1
Dzongkha	dz-BT	1
English (Australia)	en-AU	1
English (Canada)	en-CA	1
English (India)	en-IN	1
English (Ireland)	en-IE	1
English (United Kingdom)	en-GB	1
English (United States)	en-US	1
Estonian	et-EE	1
Faroese	fo-FO	1
Filipino	fil-PH	1
Finnish	fi-FI	1
French (Belgium)	fr-BE	1
French (Canada)	fr-CA	1
French (France)	fr-FR	1
French (Switzerland)	fr-CH	1

NAME	LOCALE CODE	KEYBOARD LAYOUT VALUE
Galician	gl-ES	1
Georgian	ka-GE	1
German (Germany)	de-DE	1
German (Switzerland)	de-CH	1
Greek	el-GR	1
Greenlandic	kl-GL	1
Guarani	gn-PY	1
Gujarati - INSCRIPT	gu-IN	1
Gujarati - Phonetic	gu-IN	2
Hausa	ha-Latn-NG	1
Hebrew	he-IL	1
Hindi - 37-key	hi-IN	1
Hindi - INSCRIPT	hi-IN	3
Hindi - Phonetic	hi-IN	2
Hinglish	hi-Latn	1
Hungarian	hu-HU	1
Icelandic	is-IS	1
Igbo	ig-NG	1
Indonesian	id-ID	1

NAME	LOCALE CODE	KEYBOARD LAYOUT VALUE
Inuktitut - Latin	iu-Latn-CA	1
Irish	ga-IE	1
Italian	it-IT	1
Japanese - 12-key	ja-JP	1
Japanese - QWERTY	ja-JP	2
Kannada - INSCRIPT	kn-IN	1
Kannada - Phonetic	kn-IN	2
Kazakh	kk-KZ	1
Khmer	km-KH	1
Kinyarwanda	rw-RW	1
Kiswahili	sw-KE	1
Konkani	kok-IN	1
Korean - 12-key Chunjiin	ko-KR	2
Korean - 12-key Naratgeul	ko-KR	3
Korean - 12-key Sky	ko-KR	4
Korean - QWERTY	ko-KR	1
Kyrgyz	ky-KG	1
Lao	lo-LA	1
Latvian	lv-LV	1

NAME	LOCALE CODE	KEYBOARD LAYOUT VALUE
Lithuanian	lt-LT	1
Luxembourgish	lb-LU	1
Macedonian	mk-MK	1
Malay (Brunei Darussalam)	ms-BN	1
Malay (Malaysia)	ms-MY	1
Malayalam - INSCRIPT	ml-IN	1
Malayalam - Phonetic	ml-IN	2
Maltese	mt-MT	1
Maori	mi-NZ	1
Marathi - INSCRIPT	mr-IN	1
Marathi - Phonetic	mr-IN	2
Mongolian - Cyrillic	mn-MN	1
Mongolian - Traditional Mongolian	mn-Mong-CN	1
Myanmar	my-MM	1
Nepali	ne-NP	1
Norwegian - Bokmal	nb-NO	1
Norwegian - Nynorsk	ny-NO	1
Odia - INSCRIPT	or-IN	1
Odia - Phonetic	or-IN	2

NAME	LOCALE CODE	KEYBOARD LAYOUT VALUE
Pashto	ps-AF	1
Persian	fa-IR	1
Polish	pl-PL	1
Portuguese (Brazil)	pt-BR	1
Portuguese (Portugal)	pt-PT	1
Punjabi - INSCRIPT	pa-IN	1
Punjabi - Phonetic	pa-IN	2
Romanian	ro-RO	1
Romansh	rm-CH	1
Russian	ru-RU	1
Sakha	sah-RU	1
Sami, Northern (Norway)	se-NO	1
Sami, Northern (Sweden)	se-NO	1
Scottish Gaelic	gd-GB	1
Serbian - Cyrillic	sr-Cyrl-RS	1
Serbian - Latin	sr-Latn-RS	1
Sesotho sa Leboa	nso-ZA	1
Setswana	tn-ZA	1
Sinhala	si-LK	1

NAME	LOCALE CODE	KEYBOARD LAYOUT VALUE
Slovak	sk-SK	1
Slovenian	sl-SI	1
Sorbian, Upper	hsb-DE	1
Spanish (Mexico)	es-MX	1
Spanish (Spain)	es-ES	1
Swedish	sv-SE	1
Syriac	syr-SY	1
Tajik	tg-Cyril-TJ	1
Tamazight (Central Atlas) - Tifinagh	tzm-Tfng-MA	1
Tamazight (Central Atlas) - Latin	tzm-Latn-DZ	1
Tamil - INSCRIPT	ta-IN	1
Tamil - Phonetic	ta-IN	2
Tatar	tt-RU	1
Telugu - INSCRIPT	te-IN	1
Telugu - Phonetic	te-IN	2
Thai	th-TH	1
Tibetan	bo-CN	1
Turkish	tr-TR	1
Turkmen	tk-TM	1

NAME	LOCALE CODE	KEYBOARD LAYOUT VALUE
Ukrainian	uk-UA	1
Urdu	ur-PK	1
Uyghur	ug-CN	1
Uzbek - Cyrillic	uz-Cyril-UZ	1
Uzbek - Latin	uz-Latn-UZ	1
Valencian	ca-ES-valencia	1
Vietnamese - QWERTY	vi-VN	1
Vietnamese - TELEX	vi-VN	2
Vietnamese - VNI	vi-VN	3
Welsh	cy-GB	1
Wolof	#N/A	1
Xhosa	xh-ZA	1
Yoruba	yo-NG	1
Zulu	zu-ZA	1

### Testing Steps:

1. Flash the build containing this customization to a device.
2. Go to the **keyboard** screen in **Settings**.
3. Verify that the list of keyboard languages enabled on the device is correct.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Text correction and suggestions

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners must enable text correction and text suggestions for at least one input language, and can optionally include more.

By default, the keyboard language files used for text correction and suggestions while typing are not included on the device.

## TIP

The primary audience for these topics is Original Equipment Manufacturers (OEMs). If you're a Windows device owner (consumer) and would like to learn more about power settings in Windows 10, please see results for [text correction](#) on Microsoft's community support site.

Text correction and suggestions are supported for the following input languages: Arabic, Catalan, Croatian, Czech, Danish, Dutch (Belgium), Dutch (Netherlands), English (India), English (UK), English (US), Finnish, French (Canada), French (Switzerland), French (France), German, Greek, Hebrew, Hinglish, Hungarian, Indonesian, Italian, Japanese, Korean, Malay, Norwegian (Bokmål), Persian, Polish, Portuguese (Brazil), Portuguese (Portugal), Romanian, Russian, Serbian (Latin), Serbian (Cyrillic), Simplified Chinese, Slovak, Spanish (Mexico), Spanish (Spain), Swedish, Traditional Chinese (Hong Kong SAR), Traditional Chinese (Taiwan), Turkish, Ukrainian, and Vietnamese.

## Instructions

For more information about language customizations, see the overview [Set languages and locales](#).

To modify the list of speech languages, the OEM must edit the **Keyboard** section of the OEMInput.xml file before building the device image. The following input languages are supported.

Input Language	Value to use in the OEMInput.xml file
Arabic	ar-SA
Catalan	ca-ES
Croatian	hr-HR
Czech	cs-CZ
Danish	da-DK
Dutch (Belgium)	nl-BE

Dutch (Netherlands)	nl-NL
English (India)	en-IN
English (UK)	en-GB
English (US)	en-US
Finnish	fi-FI
French (Canada)	fr-CA
French (Switzerland)	fr-CH
French (France)	fr-FR
German	de-DE
Greek	el-GR
Hebrew	he-IL
Hindi	hi-IN
Hinglish	hi-Latn
Hungarian	hu-HU
Indonesian	id-ID
Italian	it-IT
Japanese	ja-JP
Korean	ko-KR
Malay	ms-MY
Norwegian (Bokmål)	nb-NO

Persian	fa-IR
Polish	pl-PL
Portuguese (Brazil)	pt-BR
Portuguese (Portugal)	pt-PT
Romanian	ro-RO
Russian	ru-RU
Serbian (Latin)	sr-Latn-RS
Serbian (Cyrillic)	sr-Cyrl-RS
Simplified Chinese	zh-CN
Slovak	sk-SK
Spanish (Mexico)	es-MX
Spanish (Spain)	es-ES
Swedish	sv-SE
Traditional Chinese (Hong Kong SAR)	zh-HK
Traditional Chinese (Taiwan)	zh-TW
Turkish	tr-TR
Ukrainian	uk-UA
Vietnamese	vi-VN

OEMs must include at least one keyboard language. To include multiple languages, add additional **Language** entries to the **Keyboard** section of the OEMInput.xml file, as shown in the following sample.

```
<SupportedLanguages>
  <UserInterface>
    <Language>en-US</Language>
  </UserInterface>
  <Keyboard>
    <Language>en-US</Language>
    <Language>vi-VN</Language>
    <Language>de-DE</Language>
  </Keyboard>
  <Speech>
    <Language>en-US</Language>
  </Speech>
</SupportedLanguages>
```

## Testing steps

1. Flash the build containing this customization to a device.
2. Go to the **keyboard** screen in **Settings**.
3. Verify that the list of keyboard languages installed on the device is correct.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Customizations for maps

10/2/2018 • 2 minutes to read • [Edit Online](#)

Describes the customizations that you can configure for maps on the mobile device.

## In this section

TOPIC	DESCRIPTION
<a href="#">Map data on an SD card and map preload</a>	Map data is used by the Maps application and the map control for third-party applications. OEMs can choose to store this data on an SD card, which provides the advantage of saving internal memory space for user data and allows the user to download more offline map data. Microsoft recommends enabling the <b>UseExternalStorage</b> setting on phones with less than 8 GB of user storage and has an SD card slot.
<a href="#">Maps for phones shipped in China</a>	Microsoft recommends using the <a href="#">ChinaVariantWin10</a> setting instead of this legacy MCSF setting.  For a Windows mobile device shipping in China, partners must specify that the device is intended for that market by configuring <code>ChinaVariant</code> setting. When enabled, maps approved by the State Bureau of Surveying and Mapping in China are used and the maps are obtained from a server located in China.
<a href="#">Preloaded map data in the user store</a>	OEMs can choose a single map region to preload from the multiple regions that are available for OEMs to download from the Microsoft partner site(s) where the Kits are also available. OEMs can choose to store this data in the user store. Maps are grouped into regions, but there are some restrictions. For more information about these restrictions, see the accompanying instructions that are part of the map data download on the Microsoft partner site(s).
<a href="#">Temporary map data cache size</a>	When a user attempts to view map data for a location that was not preloaded or is not already installed on the phone, map data will be downloaded to dynamically render a map. This data is stored in a temporary cache that the Maps application maintains for this purpose. By default this cache is allowed to use a maximum of 128 MB of storage. For phones with a limited amount of available storage, OEMs can specify that the cache only use a maximum of 64 MB of storage. Microsoft recommends that this customization only be used for phones with a limited amount of internal storage space. Reducing the size of the online cache for Map data does not affect the size of the installed (or offline) maps.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Map data on an SD card and map preload

10/2/2018 • 2 minutes to read • [Edit Online](#)

Map data is used by the Maps application and the map control for third-party applications. OEMs can choose to store this data on an SD card, which provides the advantage of saving internal memory space for user data and allows the user to download more offline map data. Microsoft recommends enabling the **UseExternalStorage** setting on phones with less than 8 GB of user storage and has an SD card slot.

You can use **UseExternalStorage** whether or not you include an SD card with preloaded map data on the phone. If set to 1 (or Yes), the OS only allows the user to download offline maps when an SD card is present. If an SD card is not present, users can still view and cache maps, but they will not be able to download a region of offline maps until an SD card is inserted.

## Important

SD card performance can affect the quality of the Maps experience when maps are stored on the SD card. When an SD card is used, Microsoft recommends that OEMs test the Maps experience and the speed of map downloads with the specific SD card part that will be used on retail phones to determine if performance is satisfactory.

## Constraints: FirstVariationOnly

**UseExternalStorage** is a first boot configuration that can be set by the OEM. The OEM cannot change or use this setting after first boot.

## Instructions:

Microsoft recommends enabling **UseExternalStorage** on phones with less than 8 GB of user storage and has an SD card slot.

1. Create a customization answer file using the contents shown in the following code sample or use the sample `UseExternalStorage.xml` file.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="PreloadedMapData"
    Description="Use to preload map data on an SD card."
    Owner=""
    OwnerType="OEM">

    <Static>

        <!-- Enable external storage for map data on an SD card -->
        <Settings Path="Maps/Storage">
            <Setting Name="UseExternalStorage" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set the `Value` to one of the following:

VALUE	DESCRIPTION
0 or No	Map data will always be stored on the internal data partition of the device.
1 or Yes	Map data will be stored on an SD card, if present. If an SD card is not present, map data will be stored on the internal data partition of the device. Map region download will also be blocked until an SD card is present.

4. If including an SD card with the device, add the preloaded map data to the SD card. Unzip the appropriate map variant data package and copy the "**diskcache**" folder onto the SD card under the `d:\MapData` directory.

#### Note

When unzipping the appropriate map variant data package, you must use a file compression/decompression utility that preserves the file attributes and timestamps. If the utility does not preserve this information, the map(s) will be treated as invalid by the OS.

#### Testing:

1. Flash a build that contains this customization on a phone.
2. Launch the Maps application and open Settings from the application bar.
3. Click on the Download Maps button and verify that the expected region displays under downloaded maps.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Maps for phones shipped in China

10/2/2018 • 2 minutes to read • [Edit Online](#)

Microsoft recommends using the [ChinaVariantWin10](#) setting instead of this legacy MCSF setting.

For a Windows mobile device shipping in China, partners must specify that the device is intended for that market by configuring `ChinaVariant` setting. When enabled, maps approved by the State Bureau of Surveying and Mapping in China are used and the maps are obtained from a server located in China.

This customization may result in different maps, servers, or other configuration changes on the device.

## NOTE

If partners do not set the `ChinaVariant` setting to 1, partners may not ship the device in China.

## Constraints

None

## Instructions

1. Create a customization answer file using the contents shown in the following code sample or use the sample `MapsForChina.xml` file.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="MapsForChina"
    Description="Use to specify that the device is intended for shipping in China.
When enabled, maps approved by the State Bureau of Surveying and Mapping
and mapping in China are used and the maps are obtained from a server
located in China."
    Owner=""
    OwnerType="OEM">
    <Static>
        <Settings Path="Maps">
            <!-- Set to 0 or 'No' (to disable), or set to 1 or 'Yes' (to enable maps for China) -->
            <Setting Name="ChinaVariant" Value="" />
        </Settings>
    </Static>
</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set `ChinaVariant` to one of the following values:

VALUE	DESCRIPTION
1 or Yes	Maps approved by the State Bureau of Surveying and Mapping in China are used and the maps are obtained from a server located in China.

VALUE	DESCRIPTION
0 or No	Disables the feature.

## Testing

1. Flash the build containing this customization to a phone with a UICC.
2. Launch the maps application and verify that the maps used are the same as those approved by the State Bureau of Surveying and Mapping in China.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Preloaded map data in the user store

10/2/2018 • 2 minutes to read • [Edit Online](#)

Microsoft provides a set of free map data that OEMs can preload on the phone. This data is used by the Maps application and the map control for third-party applications. Microsoft recommends that OEMs make use of this customization, because it greatly improves the performance of the map experience by reducing the requirement for the dynamic download of map data. When a map is preloaded, the phone consumes less cellular data using maps, connectivity is also not required for map browsing, searching, or routing, and users do not have to download the map themselves.

OEMs can choose a single map region to preload from the multiple regions that are available for OEMs to download from the Microsoft partner site(s) where the Kits are also available. OEMs can choose to store this data in the user store. Maps are grouped into regions, but there are some restrictions. For more information about these restrictions, see the accompanying instructions that are part of the map data download on the Microsoft partner site(s).

**Constraints:** None

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample or use the sample PreloadedMapData.xml file.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="PreloadedMapData"
    Description="Use to preload map data in the user data store."
    Owner=""
    OwnerType="OEM">

    <Static>

        <!-- Specify map regions to preload in the user data store by setting Source to the
            source directory location of the map region you want to preload. -->
        <DataAssets Type="MapData">
            <DataAsset Source="C:\Path\Maps\Europe" />
            <DataAsset Source="C:\Path\Maps\Asia" />
            <!-- Add additional DataAsset elements for each map region you want to preload -->
        </DataAssets>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.
3. Set the **MapData** asset `Source` to the source directory location of the map region you want to include. For example, if `C:\Path\Maps\Europe` contains the downloaded map data that you want to preload, set `Source` to that directory.

To add additional maps, add a new **DataAsset** setting and set the source to the directory location of the map region you want to include.

## Tip

You can avoid wiping preloaded maps off the internal store on the factory line using the [ResetPhoneEx] API. For

more information, see [Resetting a phone during manufacturing](#).

**Testing:**

1. Flash a build that contains this customization on a phone.
2. Launch the Maps application and open Settings from the application bar.
3. Click on the Download Maps button and verify that the expected region displays under downloaded maps.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Temporary map data cache size

10/2/2018 • 2 minutes to read • [Edit Online](#)

When a user attempts to view map data for a location that was not preloaded or is not already installed on the phone, map data will be downloaded to dynamically render a map. This data is stored in a temporary cache that the Maps application maintains for this purpose. By default this cache is allowed to use a maximum of 128 MB of storage. For phones with a limited amount of available storage, OEMs can specify that the cache only use a maximum of 64 MB of storage. Microsoft recommends that this customization only be used for phones with a limited amount of internal storage space. Reducing the size of the online cache for Map data does not affect the size of the installed (or offline) maps.

**Constraints:** FirstVariationOnly

**Instructions:**

1. Create a customization answer file using the contents shown in the following code sample or use the sample UseSmallerCache.xml file.

```
<?xml version="1.0" encoding="utf-8" ?>

<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="UseSmallerCache"
    Description="Use to reduce the size of the online cache for Map data to a maximum
of 64 MB of storage."
    Owner=""
    OwnerType="OEM">

    <Static>
        <Settings Path="Maps/Storage">
            <Setting Name="UseSmallerCache" Value="" />
        </Settings>
    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.
3. Set `UseSmallerCache` to one of the following values:

VALUE	DESCRIPTION
1 or Yes	Reduces the size of the online cache for Map data to a maximum of 64 MB. This does not affect installed (offline) maps.
0 or No	Reverts the size of the online cache for Map data to the default maximum of 128 MB.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Customizations for notifications and quick actions

10/2/2018 • 2 minutes to read • [Edit Online](#)

Describes the customizations that you can configure for notifications and quick actions on the mobile device.

## In this section

TOPIC	DESCRIPTION
<a href="#">Add an LED notification option</a>	OEMs can configure a registry key to specify a selected notification LED as the LED notification and then add an LED notification option to the device's messaging Settings screen.
<a href="#">Configure Quick actions</a>	OEMs can change the default set of actions for each slot on the Quick actions screen in Notifications & actions.
<a href="#">CMAS Required Monthly Test</a>	OEMs can set a registry key so monthly CMAS messages can be delivered to the device.
<a href="#">Display CMAS message order</a>	Partners can configure the order in which newly received CMAS alert messages are displayed on the device.
<a href="#">Emergency notifications</a>	Partners can turn on support for various government emergency notification programs.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Add an LED notification option

10/2/2018 • 2 minutes to read • [Edit Online](#)

In Windows 10 Mobile, the notification LED on handheld devices may not turn on when a user receives a text message. To improve this user experience, OEMs can configure a registry key to specify a selected notification LED as the LED notification and then add an **LED notification** option to the device's messaging **Settings** screen.

**Constraints:** None

## Instructions:

1. You must configure the **HardwareId** and **InstanceId** registry keys to enable LED notification on the device. In the following example, you must change the value of **HardwareId** to match your device ID (DeviceId).

```
[HKEY_LOCAL_MACHINE\Microsoft\Shell\Nocontrol\LedAlert]
"HardwareId"="ACPI\QCOM0D50"
"InstanceId"=dword:0
```

**HardwareId** specifies the HardwareId for the LED while **InstanceId** specifies the InstanceId for the selected notification LED.

If the OS correctly detects the LED, the following registry keys will also be populated. Otherwise, they will not be created.

```
"LedHwAvailable"=dword:00000001
"Intensity"=dword:00000064
"Period"=dword:000007d0
"Dutycycle"=dword:0000003c
"Cyclecount"=dword:ffffffff
```

Where:

- **Intensity** - Denotes the intensity, from 0-100%
  - **Period** - Specifies the period, in milliseconds
  - **Dutycycle** - Specifies the duty cycle, from 0-100%
  - **Cyclecount** - Specifies the number of repetitions per cycle
2. To add the registry keys and their values to the OS image, create a new .pkg.xml file or modify an existing one and then add a **RegKeys** element to the .pkg.xml.

The following example shows how to do this.

```

<?xml version="1.0" encoding="utf-8"?>
<Package xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  Owner=""
  Component=""
  SubComponent=""
  OwnerType="OEM"
  ReleaseType="">
  <Components>
    <OSComponent>
      <RegKeys>
        <RegKey KeyName="$(hk1m.software)\Microsoft\Shell\Nocontrol\LedAlert">
          <RegValue
            Name="HardwareId"
            Type="REG_SZ"
            Value="ACPI\QCOM0D50"
            />
          <RegValue
            Name="InstanceId"
            Type="REG_DWORD"
            Value="0"
            />
        </RegKey>
      </RegKeys>
    </OSComponent>
  </Components>
</Package>

```

Specify the values for the **Owner**, **Component**, **SubComponent**, and **ReleaseType**. For example:

- **Owner**=“Contoso”
- **Component**=“LEDNotification”
- **SubComponent**=“EnableLEDArt”
- **ReleaseType**=“Test”

You must also replace the **Value** for **HardwareId** to one that matches your LED's DeviceId.

3. Name and save your .pkg.xml file, then generate a package (.spkg) using the .pkg.xml as input.
4. After you've created the .spkg, define the specific types of image builds that you want to contain the package.

For example, the following code snippet shows a sample OEM feature manifest (FM) file that may contain the .spkg that includes the customization:

```

<?xml version="1.0" encoding="utf-8"?>
<FeatureManifest
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate">
  <!-- Sample FM File -->
  <Features>
    <OEM>
      <PackageFile Path="SourceDirectory" Name="Contoso.LEDNotification.EnableLEDArt.spkg">
        <FeatureIDs>
          <FeatureID>WEH_LEDALERT</FeatureID>
        </FeatureIDs>
      </PackageFile>
    </OEM>
  </Features>
</FeatureManifest>

```

In this example, replace *SourceDirectory* with the location that contains the .spkg that you created in the

previous step. Also, replace the example *Contoso.LEDNotification.EnableLEDArt.spkg* with the actual name of the .spkg file. **FeatureID** specifies the ID that you're associating with the .spkg. You can provide a different name if you'd like.

- a. Once you've defined the feature, modify your OEMInput.xml file to add a **Features** element (if one doesn't already exist), add a new **OEM** child element (if one doesn't already exist), and add a new **Feature** entry with the name of the feature that you just defined.

For example, the OEMInput.xml entry for the feature you defined in the previous step will look like this:

```
<Features>
  <OEM>
    <Feature>WEH_LEDALERT</Feature>
  </OEM>
</Features>
```

For more information about OEMInput.xml, see [OEMInput file contents](#).

5. Build the OS image. For more information, see *Using ImgGen.cmd to generate an image* in [Build a mobile image using ImgGen.cmd](#).

#### Testing:

1. Flash the build that contains this customization to a mobile device.
2. Go through OOBE to set up your device.
3. Open the messaging app's **Settings** screen and verify that there's an option for **LED notification**. Make sure this option is checked or enabled.
4. From another device, send a text message to the device that has LED notification turned on. Verify that you LED notification turned on when the text message was received.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Configure Quick actions

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can change the default set of actions for each slot on the **Quick actions** screen in **Notifications & actions**.

The **Notifications & actions** settings screen contains a section at the top for users to configure **Quick actions**, which are actions that users can have available for quick access or without having to open the apps list or the settings screen to find them. Each quick action has a slot. If a user selects a quick action to go into an occupied slot (for example, Slot 1), and the chosen action already exists in another slot (for example, Slot 2), the two quick actions will swap so that the user-selected action always moves to the slot that the user has selected even though the action may already be in another slot.

## NOTE

In Windows 10 Mobile, the quick actions are not configurable through MCSF settings or Windows provisioning. OEMs must directly set the registry key to change the OS default quick actions.

Slots are ordered right-to-left so Slot 1 is always on the right and Slot 5 only appears in large screen devices.

The default pinned quick actions for 4-slot mobile devices are:

- Slot 4: Wi-Fi
- Slot 3: Bluetooth
- Slot 2: Rotation lock
- Slot 1: All settings

The default pinned quick actions for 5-slot mobile devices are:

- Slot 5: Camera
- Slot 4: Wi-Fi
- Slot 3: Bluetooth
- Slot 2: Rotation lock
- Slot 1: All settings

OEMs can change the default quick action for each slot. If an OEM chooses not to configure all the slots available for the device, only the slot that was set will be changed and the other default Windows quick actions will remain set.

A slot cannot be empty. If an OEM sets the value for Slot 5, but the mobile device is not a large screen device, the OS ignores the value set for Slot 5. If an invalid value is used, the OS ignores the setting.

## Instructions:

### To override the default pinned quick actions

- Set the `HKLM\Software\Microsoft\Shell\OEM\QuickActions\Slot` registry key and then set the value of the slot number that you want to configure to a friendly name value.

The following table shows the friendly names that you can use as the value for the slot number that you want to configure.

FRIENDLYNAME	DESCRIPTION
Microsoft.QuickAction.AllSettings	Pins All settings
Microsoft.QuickAction.Connect	Pins the Connect app
Microsoft.QuickAction.Note	Pins the Note app
Microsoft.QuickAction.Flashlight	Pins the Flashlight app
Microsoft.QuickAction.RotationLock	Pins Rotation lock
Microsoft.QuickAction.BatterySaver	Pins Battery saver
Microsoft.QuickAction.Bluetooth	Pins Bluetooth settings
Microsoft.QuickAction.WiFi	Pins Wi-Fi settings
Microsoft.QuickAction.AirplaneMode	Pins Airplane mode settings
Microsoft.QuickAction.Vpn	Pins VPN settings
Microsoft.QuickAction.Cellular	Pins Cellular settings
Microsoft.QuickAction.MobileHotspot	Pins Mobile hotspot settings
Microsoft.QuickAction.Camera	Pins the Camera app
Microsoft.QuickAction.Brightness	Pins Brightness
Microsoft.QuickAction.QuietHours	Pins Quiet hours
Microsoft.QuickAction.Location	Pins Location settings

For example, to change Slot 4 on 4-slot mobile devices from Wi-Fi to Airplane mode, set the value for `HKLM\Software\Microsoft\Shell\OEM\QuickActions\Slot\4` to `Microsoft.QuickAction.AirplaneMode`.

### Testing:

1. Flash the build that contains this customization to a mobile device.
2. Verify that the default quick action(s) that you set are showing up in the correct slot(s). For large screen devices, verify that there are 5 quick actions that are showing up instead of 4.
3. Navigate to the **Quick actions** screen in **Notifications & actions** screen and verify that the default quick settings action(s) that you set are also showing up in the correct slots.

## Related topics

[Prepare for Windows mobile development](#)

# CMAS Required Monthly Test

10/2/2018 • 2 minutes to read • [Edit Online](#)

Windows supports the Commercial Mobile Alert System (CMAS) Required Monthly Test (RMT) messages. To enable this, OEMs can set a registry key so messages can be delivered to the device. If this registry key is turned on, RMT messages will be displayed to the user in the same manner as other CMAS message types.

The alert title and message sender text used for threading will be **Required Monthly Test**.

There will be no changes made to the emergency alerts user settings on the device. RMT messages will only be configured by the OEM and users will not be able to change the setting. No user setting in Emergency alerts will have any impact on the RMT messages. For example, if the user selects **Presidential only** from the emergency messages settings, and the OEM turns on the CMAS RMT registry key, the user will still receive RMT messages.

Usually, this setting will only be turned on for test devices used by mobile operators. The CMAS RMT registry key will configure ports 4380 and 4381 although the latter is not RMT and is used by some mobile operators for testing purposes only.

**Constraints:** None

This customization supports: **per-SIM** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="ShowRequiredMonthlyTest"
    Description="Use to enable phones to receive CMAS RMT messages."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Messaging/PerSimSettings/$(__ICCID)">
            <Setting Name="ShowRequiredMonthlyTest" Value="" />
        </Settings>
    </Variant>
</ImageCustomizations>

```

2. Specify an **Owner**.
3. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
4. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
5. Set **ShowRequiredMonthlyTest** to one of the following values:

VALUE	DESCRIPTION
1 or True	Enable devices to receive CMAS RMT messages and have these show up on the device.
0 or False	Disable devices from receiving CMAS RMT messages.

#### Testing steps:

1. Flash the build containing this customization to a device.
2. The length of time may vary, but the device should now be configured to receive RMT messages. The alert

title and message text sender will show as **Required Monthly Test**.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Display CMAS message order

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can configure the order in which newly received CMAS alert messages are displayed on the device.

If the device receives at least one CMAS alert message which has not been acknowledged by the user, and another CMAS alert message arrives on the device, partners can configure the order in which the newly received alert messages are displayed on the device regardless of the service category of the alert. Users will not be able to change the display order once it has been set.

If partners do not specify a value for this customization, the default first in/first out (FIFO) display order is used.

Users will be able to acknowledge the messages in the reverse order they were received.

**Constraints:** FirstVariationOnly

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="DisplayCmasLifo"
    Description="Use to configure the order for displaying new CMAS alert messages."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Messaging/GlobalSettings">
            <Setting Name="DisplayCmasLifo" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set the `Value` to one of the following:

VALUE	DESCRIPTION
0 or 'False'	Sets a First in/first out (FIFO) message order. Users will not be able to see newer alert messages until they have dismissed the previous alert message(s).
1 or 'True'	Sets a Last in/first out (LIFO) message order. Newer alert messages will immediately appear on top of older alert messages.

## Testing Steps:

Work with your mobile operator partner to fully test this customization on their network.

Verify that the order in which CMAS alert messages are displayed on the device that contains the customization

matches the setting (FIFO or LIFO) that you have specified.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Emergency notifications

10/2/2018 • 9 minutes to read • [Edit Online](#)

Partners can turn on support for various government emergency notification programs.

Windows supports the following types of government emergency notification programs:

- **For devices sold in the United States**, partners can turn on support for the Commercial Mobile Alert Service (CMAS) if the mobile operator's network supports it. Partners can also specify the alert type defaults. CMAS is a federal program in the United States in which users can receive emergency notifications as high priority SMS messages in situations such as national emergencies, natural disasters, severe weather, and AMBER alerts.

The OS also supports handling of CMAS messages for one additional language on a separate range of CMAS channels (4383 – 4395) in compliance with the ATIS-0700013 (*Implementation Guidelines for Mobile Device Support of Multi-Language CMAS*) specification.

- **For devices sold in Japan**, partners can turn on the Earthquake & Tsunami Warning System (ETWS) if the mobile operator's network supports it.
- **For devices sold in the Netherlands**, partners can turn on the Netherlands Announcements if the mobile operator's network supports it.
- **For devices sold in Chile**, partners can enable the device to receive LAT-Alert Local Alerts if the mobile operator's network supports it.
- **For devices sold in Taiwan**, partners can enable the device to receive Taiwan Alerts.

Emergency messages are displayed as notifications at the top of the screen until they are dismissed by the user. Emergency messages do not support reply, forward or copy and paste. They can be received even in storage full conditions. Emergency messages use a distinct alert sound when they arrive as well as vibration.

Except for LAT-Alert Local Alerts, when an alert notification shows, users can dismiss the alert message or tap a **Settings** button to go to the messaging settings page where they can easily change their emergency alert settings after receiving an alert. The **Settings** button is only shown if the OS has been configured to show the emergency alert settings page. This button is shown for all types of emergency alerts where there is a user-visible settings page to let users control the alerts. If there are other alert messages, these messages will not be dismissed. The user always has to close each alert individually.

**Constraints:** None

This customization supports: **per-SIM** value

**Instructions:**

1. Modify the following answer file code sample based on the instructions given for the alert type that you want to enable.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="EmergencyNotifications"
    Description="Use to enable and configure the settings for certain government
    emergency notification programs."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <!-- Enable and configure the settings for the emergency notification program program.
            See the sections for the emergency notification program for more information about the settings
            you can enable. -->

        </Variant>
    </ImageCustomizations>

```

2. Specify an `Owner`.
3. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
4. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
5. The `EmOperatorEnabled` setting specifies the emergency alert user interface to be shown to the user. The values for specific country/region alert systems are shown in their respective sections.

However, if `EmOperatorEnabled` is not set, the default value is 0 or none.

Configure the settings for the government emergency notification program that you want to enable. Each section shows the correct `EmOperatorEnabled` value that you need to use to enable the government emergency notification program and the other settings you can use to fully configure the alert system:

<b>EMOPERATORENABLED VALUE</b>	<b>DESCRIPTION</b>
0	None

EMOPERATORENABLED VALUE	DESCRIPTION
1	CMAS Alerts
2	NL Alerts
3	ETWS Alerts
4	LAT Alerts
5	Taiwan Alerts

#### Testing steps:

Work with your mobile operator to test this customization on their network.

## US CMAS emergency notifications settings

Windows 10 Mobile supports these CMAS alert types: Presidential alerts, Extreme alerts, Severe alerts, and AMBER alerts.

Each CMAS emergency alert type has a toggle in the UI to allow the user to enable or disable the alert type. By default, all alert types are On. Note that Presidential alerts are read-only and always on.

Users' settings persist across update and restore, including updates and restores from apollo. Partners can set the defaults for these alert types but users' preferences take precedence.

#### Instructions:

1. Use the following settings to fully configure CMAS alert types:

```

<Settings Path="Messaging/PerSimSettings/$(__ICCID)/EmergencyAlertOptions">
    <Setting Name="EmOperatorEnabled" Value="1" />
    <Setting Name="CmasExtremeAlertEnabled" Value="" />
    <Setting Name="CmasSevereAlertEnabled" Value="" />
    <Setting Name="CmasAMBERAlertEnabled" Value="" />
    <Setting Name="SevereAlertDependentOnExtremeAlert" Value="" />
</Settings>

<!-- To fully configure CMAS, you must also set the per-device and per-IMSI DefaultMCC setting.
<Settings Path="CellCore/PerDevice/SMS">
    <Setting Name="DefaultMCC" Value="" />
</Settings>

<Settings Path="CellCore/PerIMSI/$(__IMSI)/SMS">
    <Setting Name="DefaultMCC" Value="" />
</Settings>
-->
```

2. Keep the `EmOperatorEnabled` value set to 1. This specifies CMAS Alerts.
3. Specify the values for `CmasExtremeAlertEnabled`, `CmasSevereAlertEnabled`, and `CmasAMBERAlertEnabled` to either 0 (Off) or 1 (On). These settings are on by default.
4. **Optional.** To meet requirements for certain mobile operators, OEMs may need to configure the `SevereAlertDependentOnExtremeAlert` setting to either 0 (off) or 1 (On). This enables the CMAS-Severe alert

switch to be dependent on the CMAS-Extreme alert shown in the **Emergency alerts** page in the messaging settings CPL. When enabled, if users turn off CMAS-Extreme alerts, CMAS-Severe alerts will also be turned off. CMAS-Severe alerts can only be toggled if CMAS-Extreme alerts are also turned on.

5. To fully configure CMAS, you must also set the per-device or per-SIM `DefaultMCC` setting. You may set the `DefaultMCC` setting for both paths.

The per-device path covers the scenario when an unexpected SIM target is inserted. In this case, the value for the per-device `DefaultMCC` will be used as the device-wide default value. The per-IMSI `DefaultMCC` value is used when a specific SIM is defined within the **Target** and a SIM matching the **Target** conditions is inserted into the device. In this case, the per-IMSI `DefaultMCC` value will override whatever value you set in the per-device configuration.

To fully allow device users to receive CMAS messages applicable to the United States, OEMs should set the `DefaultMCC` value to either 310 or 311. Do not use 001 even when performing a test SIM card scenario.

If you are migrating CMAS alert settings from previous Windows Phone releases to Windows 10 Mobile, use the following mapping as a guide for restoring users' settings:

PREVIOUS RELEASE SETTINGS	WINDOWS 10 MOBILE SETTINGS
AMBERT alerts: On	AMBER alert: On
Presidential alerts only	Presidential alerts: On Extreme alerts: Off Severe alerts: Off
Presidential and Extreme alerts	Presidential: On Extreme alerts: On Severe alerts: Off
All alerts	Presidential: On Extreme alerts: On Severe alerts: On

## Japan ETWS emergency notifications settings

Partners can turn on the Earthquake & Tsunami Warning System (ETWS) and configure an alert sound to be played with the alert message.

### Instructions:

1. Use the following settings to fully configure ETWS alert types:

```

<Settings Path="Messaging/PerSimSettings/$__ICCID">
    <!-- Use to add a custom alert sound
    <Asset Name="EtwsSound" Source="" />
    <Setting Name="EtwsSoundFileName" Value="" />
-->

    <Setting Name="EarthquakeMessageString" Value="" />
    <Setting Name="TsunamiMessageString" Value="" />
    <Setting Name="EarthquakeTsunamiMessageString" Value="" />
</Settings>

<Settings Path="Messaging/PerSimSettings/$__ICCID/EtwsSoundEnabled">
    <Setting Name="EmOperatorEnabled" Value="3" />
    <Setting Name="EtwsSoundEnabled" Value="" />
</Settings>

```

2. To use a custom alert sound:

- a. Add the custom alert sound file by adding the `EtwsSound` asset name and setting `Source` to the location and file name of the custom alert sound.
- b. Use the `EtwsSoundFileName` setting and set the value to the name of the sound file you added in the previous step.
3. The default strings for the tsunami and earthquake messages are localized. However, only one set of customized strings are allowed. If you override the default strings, the same custom strings are displayed regardless of the language set on the device.
  - To override the Primary Earthquake default message, specify the `EarthquakeMessageString` setting value. This string will be used regardless of what language is set on the device.
  - To override the Primary Tsunami default message, specify the `TsunamiMessageString` setting value. This string will be used regardless of what language is set on the device.
  - To override the Primary Tsunami and Earthquake default message, specify the `EarthquakeTsunamiMessageString` setting value. This string will be used regardless of what language is set on the device.
4. Keep the `EmOperatorEnabled` value set to 3. This specifies ETWS Alerts.
5. To play an ETWS alert sound, set `EtwsSoundEnabled` to 0 (Off) or 1 (On). If this value is not set, the default is on.

## Netherlands NL emergency notifications settings

The Netherlands NL alert system is based on the US CMAS system and the device shows a user experience similar to CMAS alerts.

**Instructions:**

1. Use the following settings to fully configure NL alert types:

```

<Settings Path="Messaging/PerSimSettings/$__ICCID/NlAlertOptions">
    <Setting Name="EmOperatorEnabled" Value="2" />
    <Setting Name="CmasExtremeAlertEnabled" Value="" />
</Settings>

```

2. Keep the `EmOperatorEnabled` value set to 2. This specifies NL Alerts.

3. Set `N12AlertEnabled` to 0 (Off) or 1 (On) to enable NL-Alert2 alerts.

`N12AlertEnabled` controls whether NL-Alert2 alerts, which are alert types that users can toggle, appear in the Netherlands CMAS settings page.

## Chile LAT-Alert Local Alerts

The LAT-Alert emergency alert system is based on the 3GPP emergency alerts and CMAS system. For devices shipping in Chile, partners must enable this customization.

### Instructions:

1. Use the following settings to fully configure Chile LAT-Alert Local Alert types:

```
<Settings Path="Messaging/PerSimSettings/$(__ICCID)/LatAlertOptions">
    <Setting Name="EmOperatorEnabled" Value="4" />
    <Setting Name="LatLocalAlertEnabled" Value="" />
</Settings>
```

2. Keep the `EmOperatorEnabled` value set to 4. This specifies LAT-Alert Local alerts.

3. Set `LatLocalAlertEnabled` to 0 (Off) or 1 (On) to enable LAT-Alert Local alerts to be received.

**Note** Users will not be able to configure nor disable LAT-Alert Local alerts through the Messaging settings screen.

## Taiwan Alerts

The Taiwan Emergency Alerts system provides the following support:

- Primary and secondary CMAS channels, including Required Monthly Test.
  - Users see primary language messages if the corresponding setting is enabled, such as Amber Alerts.
  - The secondary language is dependent on the language ID of the sent broadcast being equivalent to the system's language ID. So, if the user's device language is in English and the sent broadcast comes in through the secondary language channel with a Spanish language ID, the message will not show up. This also applies to CMAS alerts.
  - Message and notification display is the same as the US CMAS behavior.
- Taiwan Alert message
  - Cell broadcast messages are received in these channels: 911 (English) and 919 (Traditional Chinese)
  - Messages are displayed like regular cell broadcast messages, which means there are no special CMAS tone or vibration.

Support for the Taiwan Emergency Alerts is a regulatory requirement for devices shipping in Taiwan, so partners shipping devices in Taiwan must enable this customization.

Users can change the Taiwan Alert settings through the Messaging settings screen.

### Instructions:

1. Use the following settings to fully configure Taiwan Emergency Alerts:

```
<Settings Path="Messaging/PerSimSettings/$(__ICCID)/TaiwanAlertOptions">
  <Setting Name="EmOperatorEnabled" Value="5" />
  <Setting Name="TaiwanAlertEnabled" Value="" />
  <Setting Name="TaiwanPresidentialAlertEnabled" Value="" />
  <Setting Name="TaiwanEmergencyAlertEnabled" Value="" />
  <Setting Name="TaiwanRequiredMonthlyTestEnabled" Value="" />
</Settings>
```

2. Keep the `EmOperatorEnabled` value set to 5. This specifies Taiwan Alerts.

3. To receive Taiwan Alerts, set `TaiwanAlertEnabled` to 0 (False) or 1 (True).

Users will not be able to configure nor disable Taiwan alerts through the Messaging settings screen.

4. To receive alerts from the Leader of the Taiwan Area, set `TaiwanPresidentialAlertEnabled` to 0 (False) or 1 (True).

Users will not be able to configure this alert type through the Messaging settings screen.

5. To receive Taiwan Emergency Alerts, set `TaiwanEmergencyAlertEnabled` to 0 (False) or 1 (True).

6. To receive Taiwan Required Monthly Test Alerts, set `TaiwanRequiredMonthlyTestEnabled` to 0 (False) or 1 (True).

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Customizations for phone calls

10/2/2018 • 5 minutes to read • [Edit Online](#)

Provides information about customizations you can configure for the phone or dialer app including branding, visual voicemail, caller ID matching, dialer codes, and more.

## In this section

TOPIC	DESCRIPTION
<a href="#">Adjust the call duration information for CDMA calls</a>	On CDMA devices, call durations in the call history may include the time spent before the call was connected. This behavior results in an incorrect calculation of the call duration. OEMs can address this potential issue by enabling this customization, which excludes the time spent before the call was connected. When enabled, the customization will also adjust the call duration to indicate that zero time was spent if the call ends before the connection was made.
<a href="#">Always display the dialed phone number</a>	OEMs can change the default behavior so that the number that's displayed in the call screen always matches the dialed phone number even if the number that the call connected to may be different.
<a href="#">Branding for phone calls</a>	Partners can add a custom image that displays the mobile operator name or logo on the <b>Incoming call</b> screen.
<a href="#">Caller ID matching</a>	Mobile operators can modify the default mapping values specified by Microsoft using the settings in the <code>Phone/CallIDMatchOverrides</code> settings group. If mobile operators specify a number of digits that is outside the OS-supported range of 4-20, the OS defaults back to 6 digits. For any country/region that doesn't exist in the default mapping table, mobile operators can use the legacy <code>CallIDMatch</code> setting to specify the minimum number of digits to use for matching caller ID.
<a href="#">Cause codes</a>	On GSM networks, OEMs can enable mobile operator-defined codes, or cause codes, to show in the UI whenever a phone call is terminated unexpectedly.
<a href="#">Conditional call forwarding</a>	Partners can now show the call forwarding icon for conditional call forwarding as well as unconditional call forwarding.
<a href="#">Configure DTMF tones</a>	Partners can configure DTMF tone settings when VoLTE calls are supported.
<a href="#">Configure message waiting indicator notifications</a>	Depending on the scenario that partners want to support, partners can configure the voicemail system so the device either ignores or responds to message waiting indicator (MWI) notifications.

TOPIC	DESCRIPTION
Dialer codes for supplementary services	Partners can define a dialer code to use for services like changing the pin, changing the password, caller ID, call forwarding, call waiting, call blocking, and so on.
Dialer codes to launch diagnostic applications	To use an OEM diagnostic app in environments such as a service center, OEMs can configure special dialer codes to start the application. OEMs can also configure dialer codes to start apps to interact with mobile operator networks or to diagnose phone malfunctions.
Dial string overrides when roaming	Partners can map certain dial strings to corresponding override numbers that are dialed when the user is roaming. To the user, it will appear as if the original number was dialed.
Disable link to contact card in active call screen	Disable the ability to access a contact's information while in the active call screen.
Disable video upgrade Store navigation	Disable automatic navigation to the Microsoft Store when the user attempts a video upgrade during a phone call, for which there is no installed app.
Disable voicemail phone number display	Disable voicemail phone number display on the call progress screen.
Dismiss the last USSD waiting dialog	Dismiss the last USSD waiting dialog in the case where there is a sequence of USSD or SIM app dialogs.
Emergency phone numbers	Partners can edit the list of valid emergency phone numbers for the market in which the phone will be sold.
Enable call recording by default	Partners can configure devices to have the call recording feature enabled by default.
Enable IMS services	Partners can identify which IP Multimedia Subsystem (IMS) services, if any, are enabled on the device by default. The IMS services that can be identified are: IMS, SMS over IMS, Voice over IMS, and Video over IMS.
Enable RCS	OEMs can configure the RCS settings using the multivariant support in the OS. Using these settings, you can specify whether the device is RCS-enabled, specify whether to use single registration for RCS, and configure the user experience for RCS.
Hide call forwarding	Partners can hide the user option for call forwarding.
Maximum number of participants in a VoLTE conference call	OEMs can specify the maximum number of participants or callers that can be added to a voice over LTE (VoLTE) conference call based on the mobile operator's network requirements.

Topic	Description
<a href="#">Network-controlled caller ID settings</a>	For markets or mobile operators that require support for network-controlled settings for outgoing caller ID, OEMs can configure the setting to indicate whether the network default setting is allowed and specify the default initial value for the caller ID setting.
<a href="#">Override the voicemail number on the UICC</a>	Mobile operators can override the voicemail number on the UICC with a different voicemail number that is configured in the registry.
<a href="#">Supplementary services exclusions</a>	Partners can define a dialer code to use for 3GP USSD services like changing the pin, changing the password, caller ID, call forwarding, call waiting, call barring, and so on. Partners can define new mappings or disable the default mappings. To define a new mapping or change the behavior of a provided supplementary service mappings, see <a href="#">Dialer codes for supplementary services</a>
<a href="#">Trim supplementary service codes</a>	OEMs can trim supplementary service codes to ensure network compatibility. When a code is sent using a USSD string in a ##code# format, <code>EnableSupplementaryServiceEraseToDeactivateOverride</code> trims the USSD string so #code# is sent. This customization applies only to codes that use the ##code# format.
<a href="#">Use OK for USSD dialogs</a>	To meet certain market requirements or user expectations, OEMs can change the button label in USSD dialogs from <b>Close</b> (the default) to <b>OK</b> .
<a href="#">Use HD audio codec for call branding</a>	OEMs can customize call progress branding when a call is made using a specific audio codec.
<a href="#">Use voice domain for emergency call branding</a>	To meet mobile operator requirements, OEMs can enable the voice domain to decide whether to use <b>Emergency calls only</b> or <b>No service</b> in the phone UI branding.
<a href="#">Visual voicemail</a>	Visual voicemail supports both traditional voicemail (retrieved through a phone call) and visual voicemail. Users can select between traditional voicemail and visual voicemail when they first attempt to access voicemail. If the mobile operator does not support this visual voicemail implementation, the user will only see the traditional voicemail option. For mobile operators that have their own particular brand that they want to use instead of visual voicemail, partners can rebrand all instances of <b>Visual voicemail</b> in the Windows 10 Mobile UI to use the operator's brand.
<a href="#">Voice over LTE call indication</a>	Partners can add a string to the phone's call progress screen to indicate if the active call is a voice over LTE (VoLTE) call depending on whether the phone call is in high quality voice status such as when using AMR-WB codec.
<a href="#">Voicemail number for CDMA phones</a>	CDMA mobile operator partners who do not have the voicemail numbers on the device SIM can configure the voicemail number for their devices.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Adjust the call duration information for CDMA calls

10/2/2018 • 2 minutes to read • [Edit Online](#)

On CDMA devices, call durations in the call history may include the time spent before the call was connected. This behavior results in an incorrect calculation of the call duration. OEMs can address this potential issue by enabling this customization, which excludes the time spent before the call was connected. When enabled, the customization will also adjust the call duration to indicate that zero time was spent if the call ends before the connection was made.

**Constraints:** FirstVariationOnly

**Instructions:**

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="AdjustCDMACallTime"
    Description="Use to adjust the calculation of the duration for a CDMA call to
exclude the time
before the call was connected. "
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Phone/PhoneSettings">
            <Setting Name="AdjustCDMACallTime" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set the `AdjustCDMACallTime` setting to one of the following values:

VALUE	DESCRIPTION
0 or 'False'	Do not adjust the calculation of the duration for a phone call to exclude the time before the call was connected.  This is the default OS behavior.
1 or 'True'	Adjust the calculation of the duration for a phone call to exclude the time before the call was connected.

## Testing steps:

Work with your CDMA mobile operator partner to test this customization on their network.

1. Flash the build containing this customization to a device that's connected to a CDMA network.
2. Set up the device and then call a valid phone number until the call is connected.

If you set `AdjustCDMACallTime` to 1 or 'True', go to the call **History** and confirm that the duration for the phone call only includes the time spent from when the phone call was connected and until you disconnected the call.

3. Call a valid phone number, but end the call before the call can be connected.

If you set `AdjustCDMACallTime` to 1 or 'True', go to the call **History** and confirm that the duration for the phone call indicates zero or no time spent on the call.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Always display the dialed phone number

10/2/2018 • 2 minutes to read • [Edit Online](#)

By default, when a user dials a phone number and the call is connected, the number that shows up in the call screen is the phone number that the call is connected to. This connected phone number may not match the phone number that the user dialed.

OEMs can change the default behavior so that the number that's displayed in the call screen always matches the dialed phone number even if the number that the call connected to may be different.

**Constraints:** FirstVariationOnly

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="DisplayNumberAsDialed"
    Description="Use to display the outgoing phone number that was dialed rather than
the number that the call was connected to. "
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Phone/PhoneSettings">
            <Setting Name="DisplayNumberAsDialed" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set the `DisplayNumberAsDialed` setting to one of the following values:

VALUE	DESCRIPTION
0 or 'False'	Display the connected phone number. This may not match the number that was dialed. This is the default OS behavior.
1 or 'True'	Always display the phone number that was dialed even when the connected phone number is different.

## Testing steps:

Work with your mobile operator partner to test this customization on their network.

1. Flash the build containing this customization to a phone.
2. Set up the phone and then call a valid phone number until the call is connected.

3. If you set `DisplayNumberAsDialed` to 1 or 'True', try dialing your voicemail number or your subscriber number and verify that the number (as dialed) is displayed in the phone's call screen UI without the call translation to the voicemail number.
4. If you set `DisplayNumberAsDialed` to 1 or 'True', try dialing one of the supplementary codes (which look like #227 or \*227, for example) and verify that the number (as dialed) is displayed in the phone's call screen UI without the call translation to the phone number that these codes translate to.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Branding for phone calls

10/2/2018 • 3 minutes to read • [Edit Online](#)

Partners can add a custom image that displays the mobile operator name or logo on the **Incoming Call** screen. This image can be hidden when the device is roaming.

Note that the **Call Progress** screen, **Call History** screen, the 12-key dialer, and the phone tile on the Start page display the PLMN string of the mobile operator, but this is not customizable. It is possible to display error messages about the registration status on these components if an invalid UICC is inserted. Alternately, partners can choose to display a longer version of the error messages that includes a reject code.

VoIP applications can also add a custom image that displays the mobile operator name and logo to the Incoming Call screen.

The custom image must meet the following requirements:

- .PNG format
- 40 px high and no more than 180 px wide
- Transparent background
- White logo – This can contain text or an image, but keep in mind that the image is not localized, and so any text should work regardless of the display language.

**Constraints:** None

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="BrandingForPhoneCalls"
    Description="Use to add a custom image that displays the mobile operator name or
    logo on the Incoming Call screen."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Phone/BrandingInformation">
            <Asset Name="BrandingImages" Source="C:\Path\BrandingImagePhoneScreens.png" />
            <Setting Name="CellularBrandingImagePath" Value="BrandingImagePhoneScreens.png" />
            <Setting Name="BrandingFlags" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Add the custom branding image. To do this:

- a. Add an **Asset** element and set the `Name` to BrandingImages.
- b. Set the asset's **Source** value to the location and name of the custom .png file. For example:

C:\Path\Branding\ImagePhoneScreens.png

4. Set the value of `CellularBrandingImagePath` to the name of the custom .png file. For example:  
`BrandingImagePhoneScreens.png`
5. Specify when the branding image should be displayed or display registration status and reject codes in the phone UI by setting the value of `BrandingFlags`.

The default value of the `BrandingFlags` setting is 0x000000FB. This value can be replaced by a bit-wise **OR** of the following flags:

FLAG	VALUE	SET BY DEFAULT	DESCRIPTION
ShowBrandingImage	0x00000001	Yes	Display the branding image for all non-roaming phone calls.
HideBrandingImageOnRoam	0x00000002	Yes	Do not display the branding image when the phone is roaming.

FLAG	VALUE	SET BY DEFAULT	DESCRIPTION
DisplayEmergencyCallsStatus	0x00000004	No	<p>For markets in which the user must be aware that emergency calls are always permitted, partners can turn on a user notification that emergency calls are still possible even if there is no SIM, the SIM is currently blocked, or the SIM is invalid.</p> <p>Setting this flag changes the phone branding from SIM state (No SIM, Invalid SIM, PIN required) to emergency possibility state (emergency only versus no service). In the call history page, the progress and dialer where a longer string is possible, the SIM state is appended to the emergency possibility state.</p> <p>This customization replaces the "No SIM" warning on the phone tile, the call history page, and the dialer with a longer message <b>Emergency only</b> on the phone tile, and <b>No SIM – Emergency calls only</b> on the dialer and call history page.</p> <p>When the phone does not have a SIM and there is no service, the phone displays <b>No SIM – No Service</b> on the call history, call progress, and the dialer. The phone token shows <b>No service</b> in this scenario.</p>

FLAG	VALUE	SET BY DEFAULT	DESCRIPTION
ShowRegistrationStatusInCallHistory	0x00000008	Yes	Display the registration status on different parts of the phone call UI when the operator name is not available because the network cannot be accessed or the SIM is locked or missing.
ShowRegistrationStatusInCallProgress	0x00000010		
ShowRegistrationStatusInDialer	0x00000020		
ShowRegistrationStatusInIncomingCall	0x00000040		
ShowRegistrationStatusInPhoneToken	0x00000080		
			When the flags are not set, the string <b>Phone</b> is used as the branding text.
ExtendedRejectCodes	0x00000100	No	<p>Display extended reject codes in error cases such as when the UICC is not provisioned or not allowed.</p> <p>To fully enable this functionality, you must also configure <a href="#">Extended error messages for reject codes</a>.</p>

### Testing steps:

1. Flash a build containing this customization to a phone that contains a UICC, or that is otherwise equipped to receive phone calls.
2. Depending on the flags you set, verify that the branding image appears appropriately on the Incoming Call screen. The branding image is added as an overlay on top of the contact image.
3. If you set `DisplayEmergencyCallStatus`, verify that the image is visible during emergency calls.
4. Remove the UICC.
5. Depending on the flags you set, verify that the registration status appears appropriately on the Call History, Call Progress, and Incoming Call screens, as well as the dialer and the phone tile.
6. If you set `ExtendedRejectCodes`, verify that messages that are displayed to alert the user that they cannot make calls include reject codes. For information about which screens are affected, see [Extended error messages for reject codes](#).

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Caller ID matching

10/2/2018 • 9 minutes to read • [Edit Online](#)

Each country/region has a varying phone number length and style so with Windows 10 Mobile the OS maps a different minimum number of digits for matching caller ID that defaults to each country/region setting on the mobile device. This enables mobile users to get the proper settings for their preferred country/region.

In Windows 10 Mobile:

- Mobile operators can modify the default mapping values specified by Microsoft using the settings in the `Phone/CallIDMatchOverrides` settings group. For more information on how to do this, see [Overriding the OS default minimum number of digits for caller ID matching](#).

If mobile operators specify a number of digits that is outside the OS-supported range of 4-20, the OS defaults back to 6 digits.

- For any country/region that doesn't exist in the default mapping table, mobile operators can use the legacy `CallIDMatch` setting to specify the minimum number of digits to use for matching caller ID. For more information on how to do this, see [Specifying the minimum number of digits for caller ID matching for other countries/regions](#).

If OEMs do not customize this setting based on mobile operator requirements, the OS uses the current default of 6, which is the last 6 digits of the phone number.

## Note

Microsoft does not recommend changing the default value without a full and complete testing of how number matching will be affected on the phone. The testing process should include numbers with and without country codes, "+", area codes, NDD, IDD, and other dialing variations. The default value has generally been found to yield the best matching results for all various dialing options for a number.

## Caller ID matching for dual SIM phones

For dual SIM phones, each SIM will use the same logic and the same minimum number of digits to match.

## Overriding the OS default minimum number of digits for caller ID matching

The following table shows a mapping of the country/region and the default minimum number of digits (**CID**) that Windows 10 uses for caller ID matching.

For any country/region that you want to modify, note the **GEOID** for the country/region and the **CID** for the country/region. You will need the **GEOID** to identify the correct setting name to use while **CID** denotes the default OS value for the country/region.

<b>GEOID</b>	<b>COUNTRY/REGION</b>	<b>CID</b>
2	Antigua and Barbuda	6
3	Afghanistan	6
4	Algeria	8

GEOID	COUNTRY/REGION	CID
5	Azerbaijan	6
6	Albania	6
7	Armenia	6
8	Andorra	6
9	Angola	9
10	American Samoa	6
11	Argentina	7
12	Australia	8
14	Austria	8
17	Bahrain	8
18	Barbados	7
19	Botswana	7
20	Bermuda	6
21	Belgium	6
22	Bahamas, The	6
23	Bangladesh	10
24	Belize	6
25	Bosnia and Herzegovina	6
26	Bolivia	6
27	Myanmar	6
28	Benin	6
29	Belarus	6
30	Solomon Islands	6
32	Brazil	8
34	Bhutan	6

GEOID	COUNTRY/REGION	CID
35	Bulgaria	6
37	Brunei	6
38	Burundi	6
39	Canada	10
40	Cambodia	6
41	Chad	6
42	Sri Lanka	9
43	Congo	6
44	Congo (DRC)	6
45	China	11
46	Chile	8
49	Cameroon	7
50	Comoros	6
51	Colombia	10
54	Costa Rica	8
55	Central African Republic	6
56	Cuba	6
57	Cabo Verde	6
59	Cyprus	6
61	Denmark	8
62	Djibouti	6
63	Dominica	6
65	Dominican Republic	7
66	Ecuador	8
67	Egypt	7

GEOID	COUNTRY/REGION	CID
68	Ireland	6
69	Equatorial Guinea	6
70	Estonia	6
71	Eritrea	6
72	El Salvador	8
73	Ethiopia	6
75	Czech Republic	8
77	Finland	9
78	Fiji Islands	6
80	Micronesia	6
81	Faroe Islands	6
84	France	6
86	Gambia, The	6
87	Gabon	6
88	Georgia	6
89	Ghana	8
90	Gibraltar	6
91	Grenada	6
93	Greenland	6
94	Germany	7
98	Greece	8
99	Guatemala	8
100	Guinea	6
101	Guyana	6
103	Haiti	6

GEOID	COUNTRY/REGION	CID
104	Hong Kong S.A.R.	6
106	Honduras	8
108	Croatia	8
109	Hungary	8
110	Iceland	6
111	Indonesia	6
113	India	10
114	British Indian Ocean Territory	6
116	Iran	10
117	Israel	7
118	Italy	9
119	Côte d'Ivoire	6
121	Iraq	7
122	Japan	9
124	Jamaica	7
125	Jan Mayen	6
126	Jordan	7
127	Johnston Atoll	6
129	Kenya	9
130	Kyrgyzstan	6
131	North Korea	6
133	Kiribati	6
134	Korea	6
136	Kuwait	8
137	Kazakhstan	7

GEOID	COUNTRY/REGION	CID
138	Laos	6
139	Lebanon	7
140	Latvia	6
141	Lithuania	6
142	Liberia	6
143	Slovakia	8
145	Liechtenstein	6
146	Lesotho	6
147	Luxembourg	6
148	Libya	7
149	Madagascar	6
151	Macao S.A.R.	6
152	Moldova	6
154	Mongolia	6
156	Malawi	6
157	Mali	7
158	Monaco	6
159	Morocco	6
160	Mauritius	6
162	Mauritania	6
163	Malta	6
164	Oman	8
165	Maldives	6
166	Mexico	7
167	Malaysia	6

GEOID	COUNTRY/REGION	CID
168	Mozambique	9
173	Niger	6
174	Vanuatu	6
175	Nigeria	8
176	Netherlands	8
177	Norway	6
178	Nepal	10
180	Nauru	6
181	Suriname	6
182	Nicaragua	8
183	New Zealand	8
184	Palestinian Authority	7
185	Paraguay	9
187	Peru	8
190	Pakistan	9
191	Poland	9
192	Panama	8
193	Portugal	7
194	Papua New Guinea	9
195	Palau	6
196	Guinea-Bissau	6
197	Qatar	8
198	Reunion	6
199	Marshall Islands	6
200	Romania	8

GEOID	COUNTRY/REGION	CID
201	Philippines	8
202	Puerto Rico	7
203	Russia	6
204	Rwanda	6
205	Saudi Arabia	8
206	St. Pierre and Miquelon	6
207	St. Kitts and Nevis	6
208	Seychelles	6
209	South Africa	9
210	Senegal	8
212	Slovenia	6
213	Sierra Leone	6
214	San Marino	6
215	Singapore	6
216	Somalia	6
217	Spain	9
218	St. Lucia	6
219	Sudan	6
220	Svalbard	6
221	Sweden	7
222	Syria	6
223	Switzerland	6
224	United Arab Emirates	9
225	Trinidad and Tobago	6
227	Thailand	6

GEOID	COUNTRY/REGION	CID
228	Tajikistan	6
231	Tonga	6
232	Togo	6
233	Sao Tomé and Príncipe	6
234	Tunisia	7
235	Turkey	6
236	Tuvalu	6
237	Taiwan	9
238	Turkmenistan	6
239	Tanzania	9
240	Uganda	9
241	Ukraine	6
242	United Kingdom	7
244	United States	8
245	Burkina Faso	6
246	Uruguay	8
247	Uzbekistan	6
248	St. Vincent and the Grenadines	6
249	Bolivarian Republic of Venezuela	8
251	Vietnam	9
252	Virgin Islands	6
253	Vatican City	6
254	Namibia	6
258	Wake Island	6
259	Samoa	6

GEOID	COUNTRY/REGION	CID
260	Swaziland	6
261	Yemen	8
263	Zambia	9
264	Zimbabwe	6
269	Serbia and Montenegro	8
270	Montenegro	6
271	Serbia	6
273	Curaçao	6
276	South Sudan	6
300	Anguilla	6
301	Antarctica	6
302	Aruba	6
303	Ascension Island	6
304	Ashmore and Cartier Islands	6
305	Baker Island	6
306	Bouvet Islands	6
307	Cayman Islands	6
309	Christmas Island	6
310	Clipperton Island	6
311	Cocos (Keeling) Islands	6
312	Cook Islands	6
313	Coral Sea Islands	6
314	Diego Garcia	6
315	Falkland Islands (Islas Malvinas)	6
317	French Guiana	6

GEOID	COUNTRY/REGION	CID
318	French Polynesia	6
319	French Southern and Antarctic Lands	6
321	Guadeloupe	6
322	Guam	6
323	Guantanamo Bay	6
324	Guernsey	6
325	Heard Island and McDonald Islands	6
326	Howland Island	6
327	Jarvis Island	6
328	Jersey	6
329	Kingman Reef	6
330	Martinique	6
331	Mayotte	6
332	Montserrat	6
334	New Caledonia	6
335	Niue	6
336	Norfolk Island	6
337	Northern Maria Islands	6
338	Palmyra Atoll	6
339	Pitcairn Islands	6
340	Rota Island	6
341	Saipan	6
342	South Georgia and the South Sandwich Islands	6
343	St. Helena	6
346	Tinian Island	6

GEOID	COUNTRY/REGION	CID
347	Tokelau	6
348	Tristan da Cunha	6
349	Turks and Caicos Islands	6
351	Virgin Islands, British	6
352	Wallis and Futuna	6
15126	Isle of Man	6
19618	Macedonia, Former Yugoslav Republic of	8
21242	Midway Islands	6
30967	Sint Maarten (Dutch part)	6
31706	Saint Martin (French part)	6
7299303	Democratic Republic of Timor-Leste	6
10028789	Åland Islands	6
161832015	Saint Barthélemy	6
161832256	U.S. Minor Outlying Islands	6
161832258	Bonaire, Saint Eustatius and Saba	6

**Constraints:** FirstVariationOnly

**Instructions:**

To modify the Microsoft-specified minimum number of digits to use for caller ID matching for one or more countries/regions, see the following example.

1. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="CallerIDMatchingOverrides"
    Description="Use to modify the default number of digits to use for matching caller
    ID."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Phone/CallIDMatchOverrides">
            <Setting Name="2" Value="" /> <!-- Use to modify the default CID for Antigua and Barbuda -->
            <Setting Name="3" Value="" /> <!-- Use to modify the default CID for Afghanistan -->
            <Setting Name="4" Value="" /> <!-- Use to modify the default CID for Algeria -->
            <Setting Name="5" Value="" /> <!-- Use to modify the default CID for Azerbaijan -->
            <Setting Name="6" Value="" /> <!-- Use to modify the default CID for Albania -->
            <Setting Name="7" Value="" /> <!-- Use to modify the default CID for Armenia -->
            <Setting Name="8" Value="" /> <!-- Use to modify the default CID for Andorra -->
            <!-- And so on if you are modifying several values -->
        </Settings>

    </Static>

</ImageCustomizations>

```

2. Specify an `Owner`.

3. In the above mapping table, identify the `Setting Name` that corresponds to the country/region you want to modify (noted by the value in the **GEOID** column in the above table) and then set the `Value` for that country/region to a new number (different from the default **CID**) that you want to use as the minimum number of digits for caller ID matching. The new `Value` must be within the OS-supported range of 4-20.

For example, to change the default minimum caller ID matching for Antigua and Barbuda (

`Setting Name="2"` ) and the Åland Islands ( `Setting Name="10028789"` ) from the default 6 to a new minimum of 8, the following settings can be set within the MCSF customization answer file.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="CallerIDMatchingOverrides"
    Description="Use to modify the default number of digits to use for matching caller
    ID."
    Owner="Contoso"
    OwnerType="OEM">

    <Static>

        <Settings Path="Phone/CallIDMatchOverrides">
            <Setting Name="2" Value="8" /> <!-- Antigua and Barbuda -->
            <Setting Name="10028789" Value="8" /> <!-- Åland Islands -->
        </Settings>

    </Static>

</ImageCustomizations>

```

Note that this simple example doesn't show variants.

## Specifying the minimum number of digits for caller ID matching for other countries/regions

**Constraints:** FirstVariationOnly

## Instructions:

For any country/region that doesn't exist in the default mapping table, mobile operators can use the legacy `CallIDMatch` setting to specify the minimum number of digits to use for matching caller ID.

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="CallerIDMatching"
    Description="Use to configure the number of digits to use for matching caller ID."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Phone/PhoneSettings">
            <Setting Name="CallIDMatch" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set the `Value` to the number of digits used in the complete phone number. The `Value` must be within the OS-supported range of 4-20.

## Testing steps:

The full testing process should include numbers with and without country codes, "+", area codes, NDD, IDD, and other dialing variations.

1. Flash the build containing this customization to a phone that has a SIM.
2. Have a second phone with a phone number.
3. Go to the **Accounts** screen in **Settings**. Configure an email account.
4. Go to the **People** app and add a new contact with a name and the phone number of the second phone.
5. Call the phone from the second phone.
6. Verify that the caller ID is matched and the contact name is displayed.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Cause codes

10/2/2018 • 5 minutes to read • [Edit Online](#)

On GSM networks, OEMs can enable mobile operator-defined codes, or cause codes, to show in the UI whenever a phone call is terminated unexpectedly.

Cause codes indicate the reason why the call ended unexpectedly and are used by mobile operators to troubleshoot customer issues and determine the source of network problems. Note that Windows 10 Mobile does not support cause codes for CDMA networks.

An OEM must work with their mobile operator partner to support this customization. The mobile operator partner, through the OEM, must do the following:

1. Specify the cause codes that must be registered. These are the cause codes that the mobile operator is interested in and the OEM uses these codes to populate the registry. Cause codes that are not registered will not be logged and will not show up in the device's UI.
2. Provide the localized string or message to display for each specific cause code in all languages corresponding to markets or locales that the mobile network covers. Each string is limited to 255 Unicode characters. If a cause code is registered, but no localized string is provided, the device UI will only show the numerical cause code.

This customization enables the cause code to persist in the call history so that the cause code is still visible even after the device reboots. The user can also view the error in the call history UI if a cause code is available.

**Constraints:** None

**Instructions:**

1. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="CauseCodes"
    Description="Use to enable operator-defined codes, or cause codes, to show in the
    UI whenever a phone call is terminated unexpectedly."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Phone/CauseCodeRegistrationTable">
            <!-- Replace $(CAUSECODEMCCMNC) with the appropriate "MCCMNC" pairs for your mobile operator. Both
            MCC and MNC have to be three digits.
            Set the Value to the network descriptor or ID. ID values can be used more than once. -->
            <Setting Name="NetworkDescriptor/$(CAUSECODEMCCMNC)" Value="" />
            <Setting Name="NetworkDescriptor/$(CAUSECODEMCCMNC)" Value="" />
        </Settings>

        <!-- Register the cause codes for the mobile operator -->

        <!-- Define a cause code and localized strings for that cause code. -->
        <Settings Path="Phone/CauseCodeRegistrationTable/$(NETWORKDESCRIPTOR)">
            <Setting Name="CauseCode/$(CAUSECODE)/$(LOCALEID)" Value="" />
            <Setting Name="CauseCode/$(CAUSECODE)/$(LOCALEID)" Value="" />
            <Setting Name="CauseCode/$(CAUSECODE)/$(LOCALEID)" Value="" />
        </Settings>

        <!-- Define a cause code and localized strings for that cause code. -->
        <Settings Path="Phone/CauseCodeRegistrationTable/$(NETWORKDESCRIPTOR)">
            <Setting Name="CauseCode/$(CAUSECODE)/$(LOCALEID)" Value="" />
            <Setting Name="CauseCode/$(CAUSECODE)/$(LOCALEID)" Value="" />
            <Setting Name="CauseCode/$(CAUSECODE)/$(LOCALEID)" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>

```

2. Specify an `Owner`.

### 3. To register the mobile network

- In the `Phone/CauseCodeRegistrationTable` settings group, replace `$(CAUSECODEMCCMNC)` in `NetworkDescriptor/$(CAUSECODEMCCMNC)` with the *MCCMNC* pair that corresponds to a *Network\_Descriptor*.
- Use the format `MCCMNC` for `$(CAUSECODEMCCMNC)`. The MCC/MNC pairs must follow this format —each must have 3 digits specified. For example, if you have an MCC/MNC pair that corresponds to 412/89, you must store this in the registry as `412089`.

The setting value for each MCC/MNC combination will be the relative path to another setting that will contain all the cause codes information for that network.

There must be one entry for each MCC/MNC pair that is supported by the mobile operator. The same network ID or descriptor can be used for all MCC/MNC entries, or multiple IDs can be defined.

- Set *Network\_Descriptor* as the value. You can use the mobile operator ID for the *Network\_Descriptor*. The network descriptor can be arbitrarily chosen by the mobile operator. For example, for the MCC/MNC pair 412/123 and 412/125, the mobile operator can choose "MOID2" as the network descriptor. MCC/MNC pairs that have the same network descriptor or mobile operator IDs will share the same cause code.

The following example shows how to describe two networks that have MCC/MNC pairs of 412/89,

412/123, and 412/125 and network descriptors MOID1, MOID2, and MOID2, respectively:

```
<Settings Path="Phone/CauseCodeRegistrationTable">
    <Setting Name="NetworkDescriptor/412089" Value="MOID1" />
    <Setting Name="NetworkDescriptor/412123" Value="MOID2" />
    <Setting Name="NetworkDescriptor/412125" Value="MOID2" />
</Settings>
```

#### 4. To specify the cause codes and localized strings for the cause code

- a. In the `Phone/CauseCodeRegistrationTable/$(NETWORKDESCRIPTOR)` settings path, replace `$(NETWORKDESCRIPTOR)` with the network IDs or descriptors you defined in the previous section.

For example, if you have two network descriptors or mobile operator IDs called MOID1 and MOID2, define a settings group for each as shown in the following example.

```
<Settings Path="Phone/CauseCodeRegistrationTable/MOID1">
    <Setting Name="CauseCode/$(CAUSECODE)/$(LOCALEID)" Value="" />
    <Setting Name="CauseCode/$(CAUSECODE)/$(LOCALEID)" Value="" />
</Settings>

<Settings Path="Phone/CauseCodeRegistrationTable/MOID2">
    <Setting Name="CauseCode/$(CAUSECODE)/$(LOCALEID)" Value="" />
    <Setting Name="CauseCode/$(CAUSECODE)/$(LOCALEID)" Value="" />
</Settings>
```

- b. For each network group, specify the cause codes, the corresponding locale IDs, and the localized strings for the cause codes.
  - `$(CAUSECODE)` must be the decimal representation of the cause code.
  - `$(LOCALEID)` must be the locale ID identifying the locale for the cause code string.
  - `Value` must be set to the localized string to display for the cause code specific to the locale `$(LOCALEID)`.

For example, to register a cause code 1 for MOID1 and specify all the localized strings for cause code 1 for the 0409 (English (US)), 0416 (Portuguese (Brazil)), and 040A (Spanish (Spain)) locales, add the following settings and values:

```
<Settings Path="Phone/CauseCodeRegistrationTable/MOID1">
    <Setting Name="CauseCode/1/0409" Value="Dialed number is unavailable -01-" />
    <Setting Name="CauseCode/1/0416" Value="Numero discado indisponivel -01-" />
    <Setting Name="CauseCode/1/040A" Value="Marcacion no disponible -01-" />
</Settings>
```

In this example, if the device's locale is set to English (US), the OS will look for the appropriate entry in the registry and display "Dialed number is unavailable -01-". If the device's locale is set to Portuguese (Brazil), "Numero discado indisponivel -01-" will be displayed, and so on.

#### Example:

The following example customization answer file shows how to register two mobile networks (MOID1 and MOID2), specify their corresponding MCC/MNC pairs, and register a set of cause codes for both mobile networks in a limited number of locales.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="CauseCodes"
    Description="Cause codes for MOID1 and MOID2."
    Owner="Contoso"
    OwnerType="OEM">

<Static>

    <Settings Path="Phone/CauseCodeRegistrationTable">
        <Setting Name="NetworkDescriptor/412089" Value="MOID1" />
        <Setting Name="NetworkDescriptor/412123" Value="MOID2" />
        <Setting Name="NetworkDescriptor/412125" Value="MOID2" />
    </Settings>

    <!-- Register the cause codes for MOID1 -->
    <Settings Path="Phone/CauseCodeRegistrationTable/MOID1">
        <!-- Cause code 1 -->
        <Setting Name="CauseCode/1/0409" Value="Dialed number is unavailable -01-" />
        <Setting Name="CauseCode/1/0416" Value="Numero discado indisponivel -01-" />
        <Setting Name="CauseCode/1/040A" Value="Marcacion no disponible -01-" />
        <!-- Cause code 3 -->
        <Setting Name="CauseCode/3/0409" Value="Unavailable -03-" />
        <Setting Name="CauseCode/3/0416" Value="Nao disponivel -03-" />
        <Setting Name="CauseCode/3/040A" Value="Usuario no disponible -03-" />
        <!-- Cause code 8 -->
        <Setting Name="CauseCode/8/0409" Value="User busy -08-" />
        <Setting Name="CauseCode/8/0416" Value="Usuario ocupado -08-" />
        <Setting Name="CauseCode/8/040A" Value="Usuario ocupado -08-" />
    </Settings>

    <!-- Register the cause codes for MOID2 -->
    <Settings Path="Phone/CauseCodeRegistrationTable/MOID2">
        <!-- Cause code 18 -->
        <Setting Name="CauseCode/18/0409" Value="No answer -18-" />
        <Setting Name="CauseCode/18/0416" Value="Sem resposta -18-" />
        <Setting Name="CauseCode/18/040A" Value="Usuario no responde -18-" />
        <!-- Cause code 29 -->
        <Setting Name="CauseCode/29/0409" Value="Call ended -29-" />
        <Setting Name="CauseCode/29/0416" Value="Chamada terminada -29-" />
        <Setting Name="CauseCode/29/040A" Value="Llamada terminada -29-" />
    </Settings>

</Static>

</ImageCustomizations>

```

## Testing:

An OEM must work with their mobile operator partner to fully test this customization on the mobile operator's network.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Conditional call forwarding

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can now show the call forwarding icon for conditional call forwarding as well as unconditional call forwarding.

Partners should not enable this feature for networks that support voicemail, which is implemented on the network as conditional call forwarding so the call forwarding icon would also be on.

**Constraints:** None

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="ConditionalCallForwardingIcon"
    Description="Use to show the call forwarding icon for conditional call forwarding
as well as unconditional call forwarding."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Shell/SystemTray/ConditionalCallForwarding">
            <!-- Set the value to 0 or 'Disabled' (shows the icon for unconditional call forwarding only), or
                set to 1 or 'Enabled' (shows the icon for both conditional and unconditional call forwarding)
            -->
            <Setting Name="ConditionalCallForwardingIcon" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set `ConditionalCallForwardingIcon` to one of the following values:

VALUE	DESCRIPTION
0 or 'Disabled'	Only unconditional forwarding will indicate call forwarding. This is the default.
1 or 'Enabled'	Both conditional and unconditional forwarding will indicate call forwarding.

## Testing:

1. Flash the build containing this customization to a device with a UICC.
2. Set conditional call forwarding using the following USDD codes as specified in the topic [Supplementary services exclusions](#):

- 61 (FWDNOREPLY)
  - 62 (FWDNOTREACHABLE)
  - 67 (FWDBUSY)
3. Depending on the market, verify that the call forwarding icon appears based on the `IndicateConditionalCallForwarding` registry setting.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Configure DTMF tones

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can configure the following DTMF tone settings when VoLTE calls are supported:

1. Duration of the DTMF tones, and the delay, or pause, between DTMF digits.
2. Enable UX option to switch between long and short DTMF tones.
3. Enable long DTMF tones if the user presses a dialpad key for an extended period.

## Configure duration of DTMF tones, and delay between digits

**Constraints:** None This customization supports: **per-IMSI** value

### Instructions

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="DTMFtones"
    Description="Use to configure settings for DTMF tones."
    Owner=""
    OwnerType="OEM">

    <Settings Path="CellCore/PerDevice/General">
        <Setting Name="DTMFOnTime" Value="" />
        <Setting Name="DTMFOffTime" Value="" />
    </Settings>
</ImageCustomizations>
```

2. Specify an `Owner`.
3. To specify the length of time, in milliseconds, to generate the DTMF tone, set `DTMFOnTime` to a value between 64 and 1000 (inclusive). For example, a value of 160 specifies 0.16 second.
4. To specify the length of time, in milliseconds, to pause between DTMF digits, set `DTMFOffTime` to a value between 64 and 1000 (inclusive). For example, a value of 120 specifies 0.12 second.

### Testing

Work with your mobile operator to fully test this customization on their network.

1. Flash the build containing this customization to a phone that has VoLTE enabled.
2. Make a VoLTE call to a service where DTMF tones can be used.
3. Verify that DTMF tones are recognized correctly. Depending on the values you specified, verify the duration of the DTMF tone and the delay between DTMF digits.

## Enable UX option to toggle between short and long DTMF tones

Partners can make a user option visible that makes it possible to toggle between short and long DTMF tones.

By default, the user option for toggling between short and long tones is hidden on GSM phones and visible for CDMA phones.

**Constraints:** FirstVariationOnly

## Instructions

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="ShowLongTones"
    Description="Use to make the user option for toggling between short and long tones
visible to users."
    Owner=""
    OwnerType="OEM">
<Static>
    <Settings Path="Phone/PhoneSettings">
        <Setting Name="ShowLongTones" Value="" />
    </Settings>
</Static>
</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set `ShowLongTones` to one of the following values:

VALUE	DESCRIPTION
0 or False	Hides the user option to make it possible for users to toggle between short and long tones.
1 or True	Shows the user option.

## Enable long DTMF tones on long keypress

OEMs can enable long DTMF tones if the user presses a dialpad key for an extended period.

**Constraints:** FirstVariationOnly

## Instructions

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="ContinuousDTMFEEnabled"
    Description="Use to enable long DTMF tones if the user presses a dialpad key for an
extended period."
    Owner=""
    OwnerType="OEM">
<Static>
    <Settings Path="Phone/PhoneSettings">
        <Setting Name="ContinuousDTMFEEnabled" Value="" />
    </Settings>
</Static>
</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set `ContinuousDTMFEEnabled` to one of the following values:

VALUE	DESCRIPTION
0 or 'False'	Fixed length (burst mode) DTMF tones are emitted.
1 or 'True'	DTMF tones will persist as long as the user presses a corresponding dialpad key.

## Testing

1. Flash the build that contains this customization to a phone.
2. Test this customization on both VoLTE and non-VoLTE phone calls.
  - If enabled, DTMF tones will last as long as the key is being pressed.
  - If disabled, DTMF tones will be of fixed length.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Configure message waiting indicator notifications

10/2/2018 • 2 minutes to read • [Edit Online](#)

Depending on the scenario that partners want to support, partners can configure the voicemail system so the device either ignores or responds to message waiting indicator (MWI) notifications.

**Constraints:** None

This customization supports: **per-IMSI** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="IgnoreMWINotifications"
    Description="Use to configure the voicemail system so the phone either ignores or
    responds to message waiting indicator (MWI) notifications."
    Owner=""
    OwnerType="OEM">
```

```
<!-- Define the Targets -->
<Targets>
    <Target Id="">
        <TargetState>
            <Condition Name="" Value="" />
            <Condition Name="" Value="" />
        </TargetState>
    </Target>
</Targets>

<Static>
    <Settings Path="Multivariant">
        <Setting Name="Enable" Value="1" />
    </Settings>
    <Settings Path="AutoDataConfig">
        <Setting Name="Enable" Value="0" />
    </Settings>
</Static>

<!-- Specify the Variant -->
<Variant Name="">
    <TargetRefs>
        <TargetRef Id="" />
    </TargetRefs>

    <Settings Path="Phone/PerSimSettings/${__IMSI}">
        <Setting Name="IgnoreMWINotifications" Value="" />
    </Settings>
</Variant>

</ImageCustomizations>
```

```

1. Specify an **Owner**.

2. For the **per-IMSI** case:

- a. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
- b. Define settings for a **Variant**, which are applied if the associated target's conditions are met.

3. Set `IgnoreMWINotifications` to one of the following values:

| VALUE        | DESCRIPTION  |
|--------------|--|
| 0 or 'False' | MWI functions normally and the user is notified that voicemails are available. |
| 1 or 'True'  | MWI notifications are ignored by the device.                                   |

If `IgnoreMWINotifications` is not present, MWI functions normally and the user is notified when voicemails are available.

**Testing steps:**

1. Flash the build containing this customization to a device that has a UICC.
2. On another phone, call the number for the device that contains the customization. Leave a voicemail message.
3. If visual voicemail has not been set up on the device, you can verify that the customization worked by checking if the device tile shows a voicemail notification.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Dialer codes for supplementary services

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can define a dialer code to use for services like changing the pin, changing the password, caller ID, call forwarding, call waiting, call blocking, and so on. Partners can define new mappings or disable the default mappings. For more information about the default dialer codes you can redefine or disable, see [Supplementary services exclusions](#).

## Limitations and restrictions:

- The format of the supplementary service string cannot be modified.
- The data sent over the network for a given operation cannot be modified.

**Constraints:** FirstVariationOnly

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="SupplementaryServicesDialerCodes"
    Description="Use to define new mappings or disable the default mappings for dialer
    codes."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Phone/SupplementaryServiceCodeOverrides">
            <!-- Select the dialer codes to disable or redefine from the set below
            <Setting Name="002" Value="" />
            <Setting Name="004" Value="" />
            <Setting Name="03" Value="" />
            <Setting Name="04" Value="" />
            <Setting Name="042" Value="" />
            <Setting Name="052" Value="" />
            <Setting Name="070" Value="" />
            <Setting Name="071" Value="" />
            <Setting Name="072" Value="" />
            <Setting Name="073" Value="" />
            <Setting Name="074" Value="" />
            <Setting Name="075" Value="" />
            <Setting Name="076" Value="" />
            <Setting Name="077" Value="" />
            <Setting Name="078" Value="" />
            <Setting Name="079" Value="" />
            <Setting Name="21" Value="" />
            <Setting Name="30" Value="" />
            <Setting Name="300" Value="" />
            <Setting Name="31" Value="" />
            <Setting Name="33" Value="" />
            <Setting Name="330" Value="" />
            <Setting Name="331" Value="" />
            <Setting Name="332" Value="" />
            <Setting Name="333" Value="" />
            <Setting Name="35" Value="" />
            <Setting Name="351" Value="" />
            <Setting Name="353" Value="" />
            <Setting Name="354" Value="" />
```

```

<Setting Name="360" Value="" />
<Setting Name="361" Value="" />
<Setting Name="362" Value="" />
<Setting Name="363" Value="" />
<Setting Name="37" Value="" />
<Setting Name="43" Value="" />
<Setting Name="591" Value="" />
<Setting Name="592" Value="" />
<Setting Name="593" Value="" />
<Setting Name="594" Value="" />
<Setting Name="61" Value="" />
<Setting Name="62" Value="" />
<Setting Name="66" Value="" />
<Setting Name="67" Value="" />
<Setting Name="75" Value="" />
<Setting Name="750" Value="" />
<Setting Name="751" Value="" />
<Setting Name="752" Value="" />
<Setting Name="753" Value="" />
<Setting Name="754" Value="" />
<Setting Name="76" Value="" />
<Setting Name="77" Value="" />
<Setting Name="96" Value="" />
-->
</Settings>

</Static>

</ImageCustomizations>

```

2. Specify an `Owner`.

### 3. To disable a default mapping

- Select the setting name that matches the dialer code that you want to disable. For example,

`Setting Name="21"` corresponds to FWDUNCONDITIONAL or the default unconditional call forwarding code.

- Set the `value` of the dialer code that you want to disable to 0x1D or 29. This causes the dialer code to send a USSD request instead.

### 4. To define a new mapping

- Select the setting name that matches the dialer code that you want to redefine.

- Set the `value` to the desired supplementary service. The following example maps dialer code 66 to unconditional call forwarding (dialer code 21):

```
<Setting Name="66" Value="21" />
```

### Testing:

- Flash the build containing this customization to a phone.
- Tap on the keypad button in **Phone**.
- Verify modified dialer codes are handled as configured.

## Related topics

[Prepare for Windows mobile development](#)

## Customization answer file overview

# Dialer codes to launch diagnostic applications

10/2/2018 • 6 minutes to read • [Edit Online](#)

To use an OEM diagnostic app in environments such as a service center, OEMs can configure special dialer codes to start the application. OEMs can also configure dialer codes to start apps to interact with mobile operator networks or to diagnose phone malfunctions.

Specific codes entered into the dialer will start the OEM dialer app. The dialer code is passed as a parameter to the OEM's primary dialer app. The primary dialer app starts when any configured dialer code is entered in the dialer.

The dialer codes customization supports these four different app types and behaviors:

- **Windows Phone Silverlight 8.0 app** - For this type of app, the legacy customization behavior remains the same. You cannot pass \\* as a dialer code and #characters are removed from the dialer code.
- **Windows Phone Silverlight 8.1 app** - For this type of app, you can define dial strings that contain \. *The app receives the dial code through navigation arguments. Arguments are in the format "DialString=##dialer\_code\*#" (for example, "DialString=##777#").*
- **Windows Runtime app** - For Windows Runtime apps, you can define dial strings that contain \. *The app receives the dial code through navigation arguments in the URI escape encoded format. Arguments are in the format "?DialString=%23%23dialer\_code%23". The app can use System.Uri.UnescapeDataString to get the arguments to format "?DialString=##dialer\_code\*#".*
- **Universal Windows app** - The behavior for this app is the same as that of a Windows Runtime app.

The OS trims the dial string for legacy apps while it passes the dial string without modifications for Universal Windows apps.

## IMPORTANT

In your PROVXML file, you must also set the **FullyPreinstall** flag to TRUE so that the app is available to run immediate after first boot or an OS update.

**Constraints:** FirstVariationOnly

## Instructions

### First-use installation of the OEM dialer application

1. Preload the OEM dialer application. To do this, use the following code example.

```

<?xml version="1.0" encoding="utf-8" ?>
<!-- Copyright (c) Microsoft Corporation. All rights reserved. -->
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="OEMDiagnosticApp"
    Description="Sample customization XML for OEM diagnostic app."
    Owner=""
    OwnerType="OEM">
    <Static>
        <Applications>
            <Application Source="C:\Customization\ TestData\apps\SampleDiagnosticApp.xap"
                License="C:\Customization\ TestData\apps\SampleDiagnosticAppLicense.xml"
                ProvXML="C:\Customization\ TestData\apps\mpap_oemmoapps_01.provxml" />
        </Applications>
    </Static>
</ImageCustomizations>

```

In the above example:

- Specify the `Name`, `Description`, and `Owner` values.
- Replace `Source` with the location and file name of the OEM dialer application source.
- Replace `License` with the location and file name of the app's license file.
- Replace `ProvXML` with the location and file name of the app's provXML file. Note that the provXML file must have the file name pattern `MPAP_*_.provxml`.

#### NOTE

To prevent the initial phone setup process from installing the application on first boot, do not place the provXML file in the directory that the initial phone setup process checks, such as `$(runtime.commonfiles)\Provisioning\OEM`. Instead, place the provXML file in another location, such as `$(runtime.commonfiles)\Xaps`, which you can then reference from the `PartnerAppProvisioningFilePath` setting.

2. You must also configure the following customization settings so that the dialer will start the primary OEM dialer application. If you set these values, no application is launched and no messages are displayed to the user. The following configuration causes the dialer to start the primary OEM dialer application.

In the customization answer file, add the following settings:

```

<Static>
    <Settings Path="Phone/PartnerAppSupport">
        <Setting Name="PartnerAppTaskUri" Value="" />
        <Setting Name="PartnerAppProvisioningFilePath" Value="\Programs\CommonFiles\Xaps\MPAP_*_.provxml" />
        <!-- Configure these settings to add additional dialer codes that can be accepted and passed as a parameter to
            the primary OEM dialer app. You can add any number of additional diagnostic codes you want to use.
            Each code should begin with ## -->
        <Setting Name="PartnerImmediateDialStrings" Value="" />
        <Setting Name="PartnerNonImmediateDialStrings" Value="" />
    </Settings>
</Static>

```

3. In the above example, provide a value for the `PartnerAppTaskUri`, the `PartnerAppProvisioningFilePath` and include any desired immediate or non-immediate dial strings.

- a. Set `PartnerAppTaskUri` to the task URI of the launched app.
  - If you're using a Universal Windows app, you can launch a diagnostic app by setting the value of `PartnerAppTaskUri` to the AUMID of a Windows app. The AUMID is in the format similar to the

package family name + the ID of the app, for example, 24f54b1d-732e-448c-b516-15078b047964\_120xq4c4hfa14!App.

- If you're using a legacy app, you can launch a diagnostic app by setting the value of `PartnerAppTaskUri` to the app URI. The app URI is in the format `app://00000000-0000-0000-0000-000000000000/_default`. Replace `00000000-0000-0000-0000-000000000000` with your app ID, such as `13372257-1b99-1712-17e7-157fc6f8557d`.
  - The dial code parameter is URI escape encoded in order to pass # and \\* characters to Windows Phone Silverlight 8.1 apps.
- b. Set `PartnerAppProvisioningFilePath` to the path (in the mobile device) and file name of the provXML that is used to install the OEM dialer app. The path must match the destination of the app you preloaded. For example, `$(runtime.commonfiles)\Xaps map to \Programs\CommonFiles\Xaps` and must be used when specifying the `PartnerAppProvisioningFilePath` value.
- c. When the dialer code is entered, the background installation process starts and when that completes, the application will be launched. If the installation is not complete within two minutes, the application is not started.
- d. Set `PartnerImmediateDialStrings` and `PartnerNonImmediateDialStrings` to add additional dialer codes that can be accepted and passed as a parameter to the primary OEM dialer app.
- Use `PartnerImmediateDialStrings` to list dial codes that invoke the OEM app immediately without the need to press the dial button. For example, `##3282#\0##634#\0##777#\0##7820#\05555`
  - Use `PartnerNonImmediateDialStrings` to list dial codes that invoke the OEM app after pressing the dial button. For example, `##634\0##3282\0##777\0##7820`

## Testing

1. Flash the build containing this customization to a phone.
2. Launch the **Phone** app and dial any of the immediate or non-immediate dial strings that you've defined.

When the dialer code is entered, the background installation process starts and when that completes, the application will be launched. If the installation is not complete within two minutes, the application is not started.

3. Verify that the defined dial strings successfully launch the diagnostic app that you preloaded.

## Dialer codes

### Password protect diagnostic functionality

Microsoft recommends that OEMs implement an input screen that is displayed when the OEM dialer application launches diagnostic functionality. This screen can request a password that is unique to the OEM's application.

### Dialer code parameter passing

The following code example shows how to parse the dialer code parameter that was passed. The code requests the **DialString** String object out of the **navigationContext** by calling the **QueryString.TryGetValue** function.

```
string dialCode = "";if (NavigationContext.QueryString.TryGetValue("DialString", out dialCode)){    int intDialCode = int.Parse(dialCode);}
```

### Adding additional dialer codes

You can add additional dialer codes so that they will be accepted and passed as a parameter to the primary OEM dialer application. You can provide any number of additional diagnostic codes that you want to use. Each dialer code should begin with `##`.

- If the dialer string ends with `#`, the app starts immediately after the last `#` is pressed. These entries are stored in

the `PartnerImmediateDialStrings` setting. You can use multiple values by separating them with a semicolon (;). For example, ##634#;##3282#;##778#;##675#;##786#

- If the dialer code does not end with #, the app starts after the call button is pressed by the user. These are stored in the `PartnerNonImmediateDialStrings` setting. You can use multiple values by separating them with a semicolon (;). For example, ##3282##2539##33284##2539##786##778##33284

Only the defined dialer codes are passed to the partner app, which can use the code to determine what programs to load or which additional screens to display.

### Predefined dialer codes

The following table describes the predefined application dialer codes available. The <Call> in the dialer code sequences represents a press of the Call button on the phone.

DIALER CODE	NUMERIC EQUIVALENT	DESCRIPTION
##DATA#	##3282#	Displays the network information.
##MFG#	##634#	Calls the manufacturing test tool, if one has been implemented.
##RTN# or ##RTN@	##786# or ##786<Call>	"Return To New"—removes all user-specific information from NVRAM and deactivates the phone.
##MSL#	##675#	Subsidy lock.
##PST# or ##PST@	##778# or ##778<Call>	Product Support Tool—Displays the product support menu to select tasks such as deactivation.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Dial string overrides when roaming

10/2/2018 • 3 minutes to read • [Edit Online](#)

Partners can map certain dial strings to corresponding override numbers that are dialed when the user is roaming. To the user, it will appear as if the original number was dialed.

The mappings are customizable and stored in the registry. All roaming override dial strings are organized under a single registry key as name-data pairs. The OS uses the dialed string as the registry name to query and the associated data as the override number to be dialed.

## Design requirements and considerations:

- Partners must specify override numbers in full international format, including a leading +. The numbers are dialed as-is on 3GPP and the + is converted to the appropriate prefix on 3GPP2.
- Dial string translation is only performed when the phone is roaming.
- Partners may specify override numbers with `&lt;SUB&gt;` to represent the subscriber number. The first instance of `&lt;SUB&gt;` in an override number is replaced with the subscriber number. Partners cannot specify `&lt;SUB&gt;` as part of a dial string to search for.

## Constraints:

None

This customization supports: **per-IMSI** value

## Instructions:

- Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="DialStringOverrides"
    Description="Use to map certain dial strings to corresponding override numbers that
    are dialed when the user is roaming."
    Owner=""
    OwnerType="OEM">
```

```

<!-- Define the Targets -->
<Targets>
    <Target Id="">
        <TargetState>
            <Condition Name="" Value="" />
            <Condition Name="" Value="" />
        </TargetState>
    </Target>
</Targets>

<Static>
    <Settings Path="Multivariant">
        <Setting Name="Enable" Value="1" />
    </Settings>
    <Settings Path="AutoDataConfig">
        <Setting Name="Enable" Value="0" />
    </Settings>
</Static>

<!-- Specify the Variant -->
<Variant Name="">
    <TargetRefs>
        <TargetRef Id="" />
    </TargetRefs>

    <Settings Path="Phone/PerSimSettings/${__IMSI}/RoamingNumberOverrides">
        <Setting Name="DialString/${(DialString)}" Value="" />
        <Setting Name="DialString/${(DialString)}" Value="" />
        <Setting Name="DialString/${(DialString)}" Value="" />
        <Setting Name="DialString/${(DialString)}" Value="" />
    </Settings>
</Variant>

</ImageCustomizations>
```

```

1. Specify an `Owner`.

2. For the **per-IMSI** case:

a. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.

b. Define settings for a **Variant**, which are applied if the associated target's conditions are met.

3. Determine the number of dial string overrides you need to configure. For each dial string:

a. Replace `$(DialString)` in the settings name with the name of the dial string or number that the user will dial when roaming. For example, if the user will dial \*611, set the setting name to:

```
<Setting Name="DialString/*611" Value="" />
```

b. Set the setting value to the override number or number that the dial string is translated into. This corresponds to the number that is actually called. For example, if the user dials \*611 and the number that must be called whenever the user dials \*611 is +18001234567, set the value as shown in the following example:

```
<Setting Name="DialString/*611" Value="+18001234567" />
```

The following example shows how to create a mapping for two dial strings, \*611 and \*86:

```
<Settings Path="Phone/PerSimSettings$/__IMSI/RoamingNumberOverrides">
  <Setting Name="DialString/*611" Value="+18001234567" />
  <Setting Name="DialString/*86" Value="+1<SUB>" />
</Settings>
```

In the preceding example:

- \*611 is the dial string and +18001234567 is the override number. The user dials \*611 and the number that is actually called is +18001234567.
- \*86 is the dial string and +1<SUB> is the override number. The user dials \*86 and the number that is actually called is +1 and the subscriber's number.

**Testing:**

To verify this customization, the phone must be roaming internationally. Work with your mobile operator partner to test this customization on their network.

1. Flash the build containing this customization to a phone.
2. Dial the numbers that you specified and verify that numbers that are actually called are the override numbers that you specified for each dial string.

Verify that the override numbers are not shown on the phone when the dial strings are called.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Disable link to contact card in active call screen

10/2/2018 • 2 minutes to read • [Edit Online](#)

Disable the ability to access a contact's information while in the active call screen.

By default, when a user is in the active call screen, the user can tap on the contact's name or phone number while in a phone call to bring up the contact's information card. The contact information card includes information such as the contact's phone numbers, email addresses, and so on.

OEMs can disable this link in the active call screen by setting the **EnableSoftwareProximitySensorMitigation** so that the contact information is not shown while in an active call. For example, if the phone does not have a proximity sensor and the user may accidentally tap the contact name or phone number with their ear during a phone call, disabling the link may be the desired user experience.

The reminders, toasts, and the shutdown curtain also assume that the proximity sensor is covered if the **EnableSoftwareProximitySensorMitigation** setting is set. When the setting is set to 1 or 'True', the device emits a noise (for toasts and reminders if sounds for these events are not silenced), but the screen does not turn on.

**Constraints:** ImageTimeOnly

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="SoftwareProximitySensorMitigation"
    Description="Use to disable the tappable contact information in the active call
screen such as for phones
without a proximity sensor."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Phone/PhoneShellUI">
            <!-- Set the Value to 0 or 'False' (to disable, default), or set to 1 or 'True' (to enable) -->
            <Setting Name="EnableSoftwareProximitySensorMitigation" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set `EnableSoftwareProximitySensorMitigation` to one of the following values:

VALUE	DESCRIPTION
0 or 'False'	The link to the contact information in the active call screen is enabled. This is the default OS behavior.

VALUE	DESCRIPTION
1 or 'True'	<p>The link to the active contact information in the active call screen is disabled.</p> <p>If event sounds for toasts and reminders are not turned off, the device emits a noise, but the screen does not turn on.</p>

### Testing:

1. Flash the build containing this customization to a phone that has a SIM or UICC.
2. Open the **Phone** app and call one of your contacts.
3. In the active call screen, verify the following behaviors depending on the value you set for

`EnableSoftwareProximitySensorMitigation` :

- If `EnableSoftwareProximitySensorMitigation` is set to 0, verify that the link to the contact information in the active call screen is enabled. If you tap the link, verify that the contact's information card is displayed while the call is ongoing.
- If `EnableSoftwareProximitySensorMitigation` is set to 1, verify that the link to the contact information in the active call screen is disabled. If you tap the link, nothing should happen and you should remain in the active call screen.

If event sounds for toasts and reminders are not turned off, verify that the device emits a noise when a toast or reminder goes off, but the screen does not turn on.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Disable video upgrade Store navigation

10/2/2018 • 2 minutes to read • [Edit Online](#)

Disable automatic navigation to the Microsoft Store when the user attempts a video upgrade for which there is no installed app.

By default, if there are no compatible video upgrade apps installed on the phone, when a user taps the video upgrade button during a phone call, a dialog is launched and the phone will navigate to the Microsoft Store. Partners can change this behavior so that if the user taps the video upgrade button, a dialog is launched that informs the user that no video app is installed, but the phone will not navigate to the Microsoft Store directly.

**Constraints:** FirstVariationOnly

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="DisableVideoUpgradeStoreNavigation"
    Description="Use to configure whether tapping the video upgrade button will launch
    a dialog to navigate
    to the Windows Phone Store or inform the user that no app is
    installed."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Phone/PhoneSettings">
            <Setting Name="DisableVideoUpgradeStoreNavigation" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set the `DisableVideoUpgradeStoreNavigation` value to one of the following:

VALUE	DESCRIPTION
0 or 'False'	Tapping the video upgrade button launches a dialog that navigates to the Store if no compatible video upgrade apps are installed on the phone.  This is the default behavior.
1 or 'True'	Tapping the video upgrade button launches a dialog that informs the user that there is no video app is installed. The phone does not navigate to the Store.

## Testing steps:

1. Flash the build containing this customization to a phone that has a UICC or Wi-Fi connection.
2. Make sure that there are no compatible video upgrade apps installed on the phone and then open the **Phone** app and call someone.
  - If `DisableVideoUpgradeStoreNavigation` is set to 0 or 'False' (or you did not change the default OS behavior), verify that a dialog is launched and that the phone navigates to the Store.
  - If `DisableVideoUpgradeStoreNavigation` is set to 1 or 'True', verify that a dialog is launched that informs the user that no video app is installed and the phone does not automatically navigate to the Store.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Disable voicemail phone number display

10/2/2018 • 2 minutes to read • [Edit Online](#)

Disable voicemail phone number display on the call progress screen.

By default, when a user calls the voicemail number, the number dialed is displayed below the **Voicemail** label on the call progress screen. If the user enters a phone number directly using the keypad, the actual number dialed (and displayed on the call progress screen) may differ from what the user entered and may potentially cause confusion. To address possible user confusion, partners can control whether the dialed voicemail phone number is displayed below the **Voicemail** label on the call progress screen.

**Constraints:** FirstVariationOnly

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="DisableVoicemailPhoneNumberDisplay"
    Description="Use to either display or hide the voicemail phone number displayed in
the call progress screen."
    Owner=""
    OwnerType="OEM">

    <Static>
        <Settings Path="Phone/PhoneSettings">
            <Setting Name="DisableVoicemailPhoneNumberDisplay" Value="" />
        </Settings>
    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set the `DisableVoicemailPhoneNumberDisplay` setting to one of the following values:

VALUE	DESCRIPTION
0 or 'False'	Shows the phone number below the <b>Voicemail</b> label on the call progress screen. This is the default OS behavior.
1 or 'True'	Hides the phone number below the <b>Voicemail</b> label on the call progress screen.

## Testing steps:

1. Flash the build containing this customization to a phone that has a UICC or SIM.
2. Open the phone app.
3. Call the voicemail either by pressing and holding "1" from the keypad screen, tapping the voicemail icon, or calling the voicemail number directly.

4. Depending on the value that you set for `DisableVoicemailPhoneNumberDisplay`, verify that the voicemail phone number is either displayed or hidden below the **Voicemail** label on the call progress screen.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Dismiss the last USSD waiting dialog

10/2/2018 • 2 minutes to read • [Edit Online](#)

Dismiss the last USSD waiting dialog in the case where there is a sequence of USSD or SIM app dialogs.

This customization affects the behavior of USSD dialog sequencing. It dismisses the last **Waiting...** dialog in the case where there is a sequence of USSD or SIM app dialogs. OEMs may need to configure this customization in cases where there is a sequence of two or more SIM app dialogs and where the OS might display a **Waiting...** dialog indefinitely and the dialog can only be dismissed when the user taps **Cancel**.

**Constraints:** None This customization supports: **per-IMSI** value

## Instructions

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="AutoDismissUssdWaitingDialog"
    Description="Use to dismiss the last 'Waiting...' dialog in cases where there is a
sequence of USSD or SIM app dialogs."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Phone/PerSimSettings/${__IMSI}">
            <Setting Name="AutoDismissUssdWaitingDialog" Value="" />
        </Settings>

    </Variant>
</ImageCustomizations>
```

2. Specify an `Owner`.

3. For the **per-IMSI** case:

- a. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
  - b. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
4. Set `AutoDismissUssdWaitingDialog` to one of the following values:

VALUE	DESCRIPTION
0 or 'False'	The last <b>Waiting...</b> dialog won't be dismissed until the user cancels to dismiss the dialog. This may be confusing in some cases and can make the dialog appear frozen.
1 or 'True'	<p>The last <b>Waiting...</b> dialog will be automatically dismissed when the sequence of USSD or SIM app dialogs completes.</p> <p>This is the default OS behavior.</p>

## Testing

Work with your mobile operator to fully test this customization on their network.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Emergency phone numbers

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can edit the list of valid emergency phone numbers for the market in which the phone will be sold.

By default, the list includes 911, 112, 08, and 999. The number 112 is hardcoded and cannot be removed. The emergency numbers apply when the dialer restricts the user to approved emergency phone numbers, such as during initial phone setup and when the phone is locked. Partners can specify which numbers can be dialed when a SIM is present and when no SIM is present.

## **Instructions:**

The emergency phone numbers are implemented by the modem vendor. For more information about how to modify the emergency dialing behavior, see the documentation provided by the modem vendor.

# Enable call recording by default

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can configure devices to have the call recording feature enabled by default.

**Constraints:** FirstVariationOnly

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="CallRecordingOff"
    Description="Indicates if call recording is turned off. User will not see call
recording functionality when this is set to true."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Phone/PhoneSettings">
            <Setting Name=" CallRecordingOff" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set `CallRecordingOff` to one of the following values:

VALUE	DESCRIPTION
0 or 'False'	Sets the call recording management app to Voice Recorder, which turns on the call recording feature.
1 or 'True'	Sets the call recording management app to none, which turns off the call recording feature. This is the default OS Value.

## Testing steps:

1. Flash the build that contains this customization to a phone.
2. Open the **Settings** app and select **Phone**.
3. Under **Default apps**, tap **Choose apps**.
4. Under **Calling**, verify that Voice Recorder is showing under **Choose the app you want to use to manage recorded phone calls**.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Enable IMS services

10/2/2018 • 3 minutes to read • [Edit Online](#)

Partners can identify which IP Multimedia Subsystem (IMS) services, if any, are enabled on the device by default. The IMS services that can be identified are: IMS, SMs over IMS, Voice over IMS, and Video over IMS.

To allow configuration of the default values of IMS services out of the box using multivariant, the `IMSServices` setting is mapped as follows:

```
IMSServices
{
    IMS  -----> RIldmconfig_ims_test_node_status -> NV 67264 subitem 'RegConfigTestMode' (// 1
means disabling IMS and 0 means enabling it)
    SMS_OverIMS -----> RIldmconfig_smsover_ip_nw_indication -> NV 67259 subitem
'iSMSOverIPNetworkIndication'
    Voice_Over_IMS -----> [RIldmconfig_ims_voice_enabled -> NV 67348 subitem 'volte_disabled']
    Video_Over_IMS -----> [RIldmconfig_ims_video_enabled -> NV 67348 subitem 'VT calling enabled']
};
```

## Note

All values need to be set at once. For example, you cannot just set the value for `Voice_Over_IMS`. You must send a value for all. The OS applies the value to the corresponding NV item only if the value is changing.

`wpblue_gdr2` allows configuration of the OMA DM services mask (sub-item of NV 69750) separately. You can use a new setting similar to `IMSServices` called `IMSOADMService` which will be directly mapped to `RIL_IMS_NW_ENABLED_FLAGS` on the modem side. See the SoC modem documentation for more details about the flags.

```
IMSOADMService
{
    0 = NONE
    1 = OMA_DM  -----> RIL_IMS_NW_ENABLED_FLAG_PROVISION (Bit 0 - Enable(1)/Disable(0) OMA DM services)
    2 = VOICE  -----> RIL_IMS_NW_ENABLED_FLAG_VOICE (Bit 1- VoLTE enable(1)/disable(0) by OMA-DM)
    4 = VIDEO  -----> RIL_IMS_NW_ENABLED_FLAG_VIDEO (Bit 2 - VT enable(1)/disable(0) by OMA-DM)
    8 = EAB_PRESENCE -----> RIL_IMS_NW_ENABLED_FLAG_EAB (Bit 3 - Presence enable(1)/disable(0) by OMA-DM)
    15 = Enable all above services
}
```

All the other settings for VoLTE and VT, such as `ShowVoLTEToggle`, `SwitchIMS`, and so on remain unchanged. For more information about these settings, see [Settings for IMS services](#).

**Constraints:** None

This customization supports: **per-IMSI** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="EnableIMSServices"
    Description="Use to configure which IMS services are enabled."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="CellCore/PerIMSI/$(__IMSI)/VoLTE">
            <Setting Name="IMSServices" Value="" />

            <!-- To configure the OMA DM services mask. -->
            <Setting Name="IMSOMADMServices" Value="" />
        </Settings>
    </Variant>
</ImageCustomizations>

```

2. Specify an `Owner`.

3. For the **per-IMSI** case:

- Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's Mobile Country Code (MCC), Mobile Network Code (MNC), and Service Provider Name (SPN).
- Define the settings for a **Variant**, which are applied if the associated target's conditions are met.

4. Set the value for the `IMSServices` setting to any combination of the following flags or bitmasks:

SERVICE	FLAG (DECIMAL)	BITMASK (BINARY)
IMS	1	0001
SMS over IMS	2	0010

SERVICE	FLAG (DECIMAL)	BITMASK (BINARY)
Voice over IMS	4	0100
Video over IMS	8	1000

You can set `IMSServices` to any decimal value formed by a combination of the bitmasks. For example, a bitmask of 1111 (or a decimal value of 15) means that all services are enabled. A bitmask of 0101 (or a decimal value of 5) means that IMS and Voice over IMS are enabled by default and SMS over IMS and Video over IMS are disabled, and so on.

1. To configure the OMA DM services mask, set the `IMSOMADMServices` setting to one of the following values:

SERVICE	FLAG (DECIMAL)	BITMASK (BINARY)
None	0	00000
OMA DM	1	00001
Voice	2	00010
Video	4	00100
EAB presence	8	01000
Enable all services	15	10000

### Testing:

Work with your mobile operator partner to test this customization on the operator's network.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Enable RCS

10/2/2018 • 6 minutes to read • [Edit Online](#)

OEMs can configure the RCS settings using the multivariant support in the OS. Using these settings, you can:

- Specify whether the device is RCS-enabled.
- Specify whether to use single registration for RCS.
- Configure the user experience for RCS.

The following design principles for RCS settings apply in Windows 10 Mobile:

- An OEM can set a policy that cannot be overwritten by the user.
- A user can set the value for a setting unless the setting is hidden by the mobile operator or OEM, or if the setting is available only to the mobile operator or OEM.
- The IMS and RCS services have a defined default behavior in the event that a policy or setting is not set.
- Backup and restore are slot-based. Any per-slot SIM settings are backups for the associated slot. When the settings are restored, they are restored in the corresponding slot even if a different SIM is in that slot.
- When there are no per-user or per-slot settings, then settings are applied per-SIM, not per-slot. For example, if the user sets group text ON for their Contoso SIM in Slot 1, and has group text OFF for their Fabrikam SIM in Slot 2, if the user swaps the Contoso SIM into Slot 2 and reboots the device, group text will be set to ON.

## RCS settings model

- **Global policy** - The global policy reads from the Windows 10 Mobile registry location and if a value isn't found, the Windows Phone 8.1 registry location is used. If no value is found in the Windows Phone 8.1 location, the messaging app uses the default behavior of the app.
- **Per-SIM policy** - The per-SIM policy is written to the Slot 1 or Slot 2 location based on the corresponding slot for the specific SIM.
  - The per-SIM policy in Slot 1 reads from the Windows 10 Mobile registry location for Slot 1. If a value is not found, the messaging app falls back to the Windows Phone 8.1 location. If no value is found in Windows Phone 8.1 location, the messaging app uses the default behavior of the app.
  - The per-SIM policy in Slot 2 reads from the Windows 10 Mobile registry location for Slot 2. If a value is not found, the messaging app uses the default app behavior.
- **Per-SIM path policy** - The per-SIM path policy is written to the Slot 1 or Slot 2 location based on the corresponding slot for the specific SIM.
  - The per-SIM policy in Slot 1 reads from the Windows 10 Mobile registry location for Slot 1. If a value is not found, the messaging app uses the default behavior of the app.
  - The per-SIM policy in Slot 2 reads from the Windows 10 Mobile registry location for Slot 2. If a value is not found, the messaging app uses the default app behavior of the app.
- **Per-provider SIM settings** - The per-provider SIM settings apply for single and dual SIM devices. The per-provider SIM settings are written to the Slot 1 or Slot 2 location based on the corresponding slot for the specific SIM. Each per-provider SIM setting (such as group text) has three separate values that determine its behavior in Windows 10 Mobile.

The following table shows the expected behavior if all of the values are set in the Windows 10 Mobile location. This applies to both Slot 1 and Slot 2. In summary, if the setting is hidden from the user, any user

setting value is ignored when the messaging app is determining which value to use.

OEM: IS IT HIDDEN?	USER SETTING VALUE	OEM DEFAULT VALUE	FINAL VALUE
No	N/A	Off	Off
No	On	Off	On
Yes	N/A	Off	Off
Yes	On	Off	Off

Per-provider SIM Slot 2 settings will fall back to the default service behavior. The following table shows the expected behavior.

WINDOWS 10 MOBILE OEM CONFIGURATION			WINDOWS 10 MOBILE BEHAVIOR		WINDOWS 10 MOBILE FINAL VALUE
OEM: IS IT HIDDEN?	USER SETTING VALUE	OEM DEFAULT VALUE	OEM: IS IT HIDDEN?	OEM AND USER DEFAULT FALBACK BEHAVIOR	
N/A	N/A	N/A	Yes	On	On
No	Off	On	Yes	On	Off
Yes	N/A	Off	Yes	On	Off
No	N/A	Off	Yes	On	Off

**Constraints:** None

This customization supports: **per-SIM** value

#### Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="EnableRCS"
    Description="Use to configure RCS settings."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>

        <!-- Configure these global settings to specify whether the device supports RCS. -->
        <Settings Path="CellCore/PerDevice/RCS">
            <Setting Name="SystemEnabled" Value="" />
            <Setting Name="UserEnabled" Value="" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <!-- Use these settings to specify whether to use single registration for RCS. -->
        <Settings Path="CellCore/PerIMSI/$(__IMSI)/RCS">
            <Setting Name="UseSingleRegistration" Value="" />
        </Settings>

        <!-- Use these settings to configure the user experience for RCS -->
        <Settings Path="Messaging/PerSimSettings/$(__ICCID)/RcsOptions">
            <Setting Name="ShowRcsEnabled" Value="" />
            <Setting Name="RcsEnabled" Value="" />
            <Setting Name="RcsSendReadReceipt" Value="" />
            <Setting Name="RcsFileTransferAutoAccept" Value="" />
        </Settings>

    </Variant>
</ImageCustomizations>

```

2. Specify an `Owner`.
3. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's Mobile Country Code (MCC), Mobile Network Code (MNC), and Service Provider Name (SPN).
4. Define the settings for a **Variant**, which are applied if the associated target's conditions are met.
5. To specify whether the system is RCS-enabled and the RCS package is installed, set `SystemEnabled` to one of the following values.

VALUE	DESCRIPTION
0 or 'No'	The system is not RCS-enabled.
1 or 'Yes'	The system is RCS-enabled. If the system supports RCS, you can also specify whether to show the user setting by configuring the value for <code>UserEnabled</code> .

6. To show the user setting if RCS is enabled on the device, set `UserEnabled` to one of the following values.

VALUE	DESCRIPTION
0 or 'No'	Don't show the user setting if RCS is enabled on the device.
1 or 'Yes'	Show the user setting if RCS is enabled on the device.

7. To specify whether to use single registration for RCS, set `UseSingleRegistration` to one of the following values.

VALUE	DESCRIPTION
0 or 'False'	Do not use single registration for RCS.
1 or 'True'	Use single registration for RCS. The RCS stack will use the modem interface to communicate with the RCS backend.

8. To configure the user experience for RCS, set the following settings.

- To show or hide the toggle for RCS activation, set `ShowRcsEnabled` to one of the following values.

VALUE	DESCRIPTION
0 or 'False'	Hides the toggle for RCS activation. This is the default OS value.
1 or 'True'	Shows the toggle for RCS activation. If you use this value, you can also configure the default value for the service by setting <code>RcsEnabled</code> .

- To set the default value for the RCS service toggle, set `RcsEnabled` to one of the following values.

VALUE	DESCRIPTION
0 or 'False'	RCS service toggle is set to Off. This is the default OS value.
1 or 'True'	RCS service toggle is set to On.

- To specify whether a read receipt is sent to the sender, set `RcsSendReadReceipt` to one of the following values.

VALUE	DESCRIPTION
0 or 'False'	A read receipt is not sent to the sender.
1 or 'True'	A read receipt is sent to the sender. This is the default OS value.

- To specify whether to automatically download an incoming RCS file transfer when the file size is less than the limit for the warning file size, set `RcsFileTransferAutoAccept` to one of the following values.

VALUE	DESCRIPTION
0 or 'False'	Do not automatically download the incoming RCS file transfer.
1 or 'True'	Do automatically download the incoming RCS file transfer. This is the default OS value.

**Testing:**

Work with your mobile operator partner to test this customization on the operator's network.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Hide call forwarding

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can hide the user option for call forwarding.

By default, users can decide whether to turn on call forwarding. Partners can hide this user option so that call forwarding is permanently disabled.

**Constraints:** FirstVariationOnly

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="HideCallForwarding"
    Description="Use to hide user option for call forwarding to users."
    Owner=""
    OwnerType="OEM">
<Static>
    <Settings Path="Phone/PhoneSettings">
        <Setting Name="HideCallForwarding" Value="" />
    </Settings>
</Static>
</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set `HideCallForwarding` to one of the following values:

VALUE	DESCRIPTION
0 or False	Shows the user option to make it possible for users to forward calls.
1 or True	Hides the user option

By default, the hide call forwarding UI is set to 0 or always shown.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Maximum number of participants in a VoLTE conference call

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can specify the maximum number of participants or callers that can be added to a voice over LTE (VoLTE) conference call based on the mobile operator's network requirements.

By default, Windows 10 Mobile supports up to 6-way conference (host + 5 participants) for VoLTE conference calls.

**Constraints:** FirstVariationOnly

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="VoLTEMaxConferenceCallPartyCount"
    Description="Use to set the maximum number of participants in a voice over LTE
conference call."
    Owner=""
    OwnerType="OEM">

    <Static>
        <Settings Path="Phone/PhoneSettings">
            <Setting Name="ConferenceCallMaximumPartyCount" Value="" />
        </Settings>
    </Static>
</ImageCustomizations>
```

2. Specify an `Owner`.
3. Set the value of `ConferenceCallMaximumPartyCount` the maximum number of participants, including the host, in a VoLTE conference call. Specify the number in decimal or hexadecimal (with the 0x prefix).

The default OS value is 6.

## Testing steps:

1. Flash the build containing this customization to a phone.
2. Work with your mobile operator partner to test this customization on their network.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Network-controlled caller ID settings

10/2/2018 • 2 minutes to read • [Edit Online](#)

For markets or mobile operators that require support for network-controlled settings for outgoing caller ID, OEMs can configure the setting to indicate whether the network default setting is allowed and specify the default initial value for the caller ID setting.

**Constraints:** None

This customization supports: **per-IMSI** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample or use the sample NetworkCallerIDSettings.xml file.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="NetworkCallerIDSettings"
    Description="Use to enable network-controlled settings for outgoing caller ID."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Phone/PerSimSettings/${__IMSI}">
            <Setting Name="ShowCallerIdNetworkDefaultSetting" Value="" />
            <Setting Name="DefaultCallerIdSetting" Value="" />
        </Settings>

    </Variant>
</ImageCustomizations>
```

2. Specify an **Owner**.
3. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and

SPN.

4. Define settings for a **Variant**, which are applied if the associated target's conditions are met.

5. To indicate whether the network default setting is allowed for the outgoing caller ID, set

`ShowCallerIdNetworkDefaultSetting` to one of the following values:

VALUE	DESCRIPTION
0 or 'False'	Not allowed. The network default option will not be shown. This is the default OS value.
1 or 'True'	Allowed. The network default option will be shown.

6. To specify the default value for the caller ID setting, set `DefaultCallerIdSetting` to one of the following values:

VALUE	DESCRIPTION
1	The caller ID is not shown for any calls.
2	The caller ID is shown only to phone contacts.
3	The caller ID is shown for all calls. This is the default OS value.
4	The network default setting is shown. If this value is chosen, OEMs must also set <code>ShowCallerIdNetworkDefaultSetting</code> to 1 or 'True'.

#### Testing steps:

Work with your mobile operator to fully test this customization on their network and verify that each setting and value behave as documented in this topic.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Override the voicemail number on the UICC

10/2/2018 • 2 minutes to read • [Edit Online](#)

Mobile operators can override the voicemail number on the UICC with a different voicemail number that is configured in the registry.

This customization can only be applied in a runtime configuration image.

**Constraints:** None

This customization supports: **per-IMSI** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>

<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="SimOverrideVoicemailNumber"
    Description="Use to set the phone to ignore the time received from an LTE network."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Phone/PerSimSettings/${__IMSI}/Critical">
            <Setting Name="SimOverrideVoicemailNumber" Value="" />
        </Settings>

    </Variant>
</ImageCustomizations>
```

2. Specify an `Owner`.

3. For the **per-IMSI** case:

- a. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
  - b. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
4. Set `SimOverrideVoicemailNumber` to a string that contains the digits of the voicemail number to use instead of the voicemail number on the UICC.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Supplementary services exclusions

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can define a dialer code to use for 3GP USSD services like changing the pin, changing the password, caller ID, call forwarding, call waiting, call barring, and so on. Partners can define new mappings or disable the default mappings. To define a new mapping or change the behavior of a provided supplementary service mappings, see [Dialer codes for supplementary services](#)

Microsoft provides a number of predefined USSD codes. Partners can exclude predefined USSD entries, allowing the number to be sent as standard DTMF tones instead. This allows for customization for specific markets where the predefined USSD codes need to be sent as a DTMF tones.

## Note

Only existing USSD codes can be overridden.

The following USSD codes are predefined in Windows Phone, and all of them can be overridden by the OEM.

CODES	DESCRIPTION	DWORD VALUE
04	CHANGEPIN	000000F4
042	CHANGEPIN2	00000F42
05	UNBLOCKPIN	000000F5
052	UNBLOCKPIN2	00000F52
03	SSCHANGEPASSWORD	000000F3
75	EMLPPBASE	00000075
750	EMLPPLVEL0	00000750
751	EMLPPLVEL1	00000751
752	EMLPPLVEL2	00000752
753	EMLPPLVEL3	00000753
754	EMLPPLVEL4	00000754
66	CALLDEFLECT	00000066

<b>CODES</b>	<b>DESCRIPTION</b>	<b>DWORD VALUE</b>
30	CALLIDCLIP	00000030
31	CALLIDCLIR	00000031
76	CALLIDCOLP	00000076
77	CALLIDCOLR	00000077
21	FWDUNCONDITIONAL	00000021
67	FWDBUSY	00000067
61	FWDNOREPLY	00000061
62	FWDNOTREACHABLE	00000062
002	FWDALL	0000FF2
004	FWDALLCONDITIONAL	0000FF4
43	CALLWAITING	00000043
360	UUSALL	00000360
361	UUSSERVICE1	00000361
362	UUSSERVICE2	00000362
363	UUSSERVICE3	00000363
33	BARROUT	00000033
331	BARROUTINTL	00000331
332	BARROUTINTLEXTOHOME	00000332
35	BARRIN	00000035

<b>CODES</b>	<b>DESCRIPTION</b>	<b>DWORD VALUE</b>
351	BARRINROAM	00000351
330	BARRALL	00000330
333	BARRALLOUT	00000333
353	BARRALLIN	00000353
354	BARRINCOMINGINTERMEDIATE	00000354
96	CALLTRANSFER	00000096
37	CALLCOMPLETEBUSY	00000037
070	PNP0	00000F70
071	PNP1	00000F71
072	PNP2	00000F72
073	PNP3	00000F73
074	PNP4	00000F74
075	PNP5	00000F75
076	PNP6	00000F76
077	PNP7	00000F77
078	PNP8	00000F78
079	PNP9	00000F79
300	CALLCNAP	00000300
591	MSP1	00000591

CODES	DESCRIPTION	DWORD VALUE
592	MSP2	00000592
593	MSP3	00000593
594	MSP4	00000594

**Constraints:** None

This customization supports: **per-IMSI** value

**Instructions:**

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="SupplementaryServicesExclusions"
    Description="Use to exclude predefined USSD entries to allow the number to be sent
as standart DTMF tones instead."
    Owner=""
    OwnerType="OEM">
```

```
<!-- Define the Targets -->
<Targets>
    <Target Id="">
        <TargetState>
            <Condition Name="" Value="" />
            <Condition Name="" Value="" />
        </TargetState>
    </Target>
</Targets>

<Static>
    <Settings Path="Multivariant">
        <Setting Name="Enable" Value="1" />
    </Settings>
    <Settings Path="AutoDataConfig">
        <Setting Name="Enable" Value="0" />
    </Settings>
</Static>

<!-- Specify the Variant -->
<Variant Name="">
    <TargetRefs>
        <TargetRef Id="" />
    </TargetRefs>

    <Settings Path="Phone/PerSimSettings/${__IMSI}">
        <Setting Name="IgnoreUssdExclusions" Value="" />
        <Setting Name="UssdExclusionList" Value="" />
    </Settings>
</Variant>

</ImageCustomizations>
```

```

1. Specify an `Owner`.

2. For the **per-IMSI** case:

a. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.

b. Define settings for a **Variant**, which are applied if the associated target's conditions are met.

3. Set `IgnoreUssdExclusions` to one of the following values:

| VALUE        | DESCRIPTION                      |
|--------------|----------------------------------|
| 0 or 'False' | Uses the USSD exclusion list.    |
| 1 or 'True'  | Ignores the USSD exclusion list. |

4. Set `UssdExclusionList` to the list of desired exclusions, separated by semicolons. For example, to override 2 and 4, set the value to `2;4`

Leading zeros are specified by using `F`. For example, to override code 079, set the value to `F79`.

### Testing:

1. Flash the build containing this customization to a phone.

2. Tap on the keypad button in **Phone**.

3. Verify modified dialer codes are handled as configured.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Trim supplementary service codes

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can trim supplementary service codes to ensure network compatibility. When a code is sent using a USSD string in a ##code# format, `EnableSupplementaryServiceEraseToDeactivateOverride` trims the USSD string so #code# is sent. This customization applies only to codes that use the ##code# format.

**Constraints:** None

This customization supports: **per-IMSI** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name=EnableSupplementaryServiceEraseToDeactivateOverride"
    Description="Enables trimming of supplementary service that use the ##code# format"
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Phone/PerSimSettings/${__IMSI}">
            <Setting Name="EnableSupplementaryServiceEraseToDeactivateOverride" Value="" />
        </Settings>
    </Variant>
</ImageCustomizations>
```

2. Specify an `Owner`.

3. For the **per-IMSI** case:

- a. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
  - b. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
4. Set `EnableSupplementaryServiceEraseToDeactivateOverride`. The possible values are:

VALUE	DESCRIPTION
0 or 'False'	Preserves codes with no trimming. This is the default OS value.
1 or 'True'	Enables trimming.

#### Testing steps:

1. Flash the build containing the customization to a phone.
2. Send a USSD code. For example, ##21# to disable conditional call forwarding.
3. Verify that conditional call forwarding has been disabled.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Use OK for USSD dialogs

10/2/2018 • 2 minutes to read • [Edit Online](#)

To meet certain market requirements or user expectations, OEMs can change the button label in USSD dialogs from **Close** (the default) to **OK**.

**Constraints:** FirstVariationOnly

**Instructions:**

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="UseOKForUssdDialogs"
    Description="Use to change the button label in USSD dialogs from 'Close' to 'OK'."
    Owner=""
    OwnerType="OEM">

    <Static>
        <Settings Path="Phone/PhoneSettings">
            <Setting Name="UseOKForUssdDialogs" Value="" />
        </Settings>
    </Static>
</ImageCustomizations>
```

2. Specify an **Owner**.

3. Change the button label in the USSD dialog by setting **UseOKForUssdDialogs** to one of the following values:

VALUE	DESCRIPTION
0 or 'False'	USSD success and failure dialogs have a single button to close the dialog labeled <b>Close</b>
1 or 'True'	USSD success and failure dialogs have a single button to close the dialog labeled <b>OK</b>

**Testing steps:**

1. Flash the build containing this customization to a phone that has a UICC.
2. Open the **Phone** app and tap the keypad button.
3. Use several USSD codes and strings to bring up a USSD success dialog and a USSD failure dialog.
4. Verify that the button label in the dialog shows **OK**.

## Related topics

[Prepare for Windows mobile development](#)

## Customization answer file overview

# Use HD audio codec for call branding

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can customize call progress branding when a call is made using a specific audio codec.

**Constraints:** None

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="EnableSupplementaryServiceEraseToDeactivateOverride" Description="Call
    progress branding"
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>
        <Settings Path="Phone/PerSimSettings/$(__IMSI)/HDAudio">
            <Setting Name="" Value="" /> <!-- Use to identify codec and string -->
        </Settings>
    </Variant>
    </ImageCustomizations>
```

2. Specify an `Owner`.
3. For the per-IMSI case:
  - a. Define Targets or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
  - b. Define Targets or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
4. In the table below, identify the `Setting Name` that corresponds to your HD audio codec, and set that as

`Setting Name`. Then, set `value` to the text string you want to use for call progress branding. For example, if you are using the EVRC audio codec, and you would like to display the text "EVRC" when using that codec, you would enter EVRCAudioQualityString in `Setting Name`, and EVRC in `value`.

**Note:** Text strings can be a maximum of 10 characters.

SETTING NAME	VALUE	DESCRIPTION
EVRCAudioQualityString	Any text string	Call progress branding for calls using the EVRC audio codec
EVRCBAudioQualityString	Any text string	Call progress branding for calls using the EVRCB audio codec
EVRCNWAudioQualityString	Any text string	Call progress branding for calls using the EVRCNW audio codec
EVRCWBAudioQualityString	Any text string	Call progress branding for calls using the EVRCWB audio codec
EVSFBAudioQualityString	Any text string	Call progress branding for calls using the EVSFB audio codec
EVSNBAudioQualityString	Any text string	Call progress branding for calls using the EVSNB audio codec
EVSSWBAudioQualityString	Any text string	Call progress branding for calls using the EVSSWB audio codec
EVSWBAudioQualityString	Any text string	Call progress branding for calls using the EVSWB audio codec
GSMEFRAudioQualityString	Any text string	Call progress branding for calls using the GSMEFR audio codec
GSMFRAudioQualityString	Any text string	Call progress branding for calls using the GSMFR audio codec
GSMFRAudioQualityString	Any text string	Call progress branding for calls using the GSMFR audio codec
QCELP13KAudioQualityString	Any text string	Call progress branding for calls using the QCELP13K audio codec

### Testing steps:

1. Flash a build containing this customization to a phone.
2. Make a phone call that uses HD audio.
3. Verify that the call progress branding is displayed during the phone call.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Use voice domain for emergency call branding

10/2/2018 • 2 minutes to read • [Edit Online](#)

To meet mobile operator requirements, OEMs can enable the voice domain to decide whether to use **Emergency calls only** or **No service** in the phone UI branding.

**Constraints:** FirstVariationOnly

**Instructions:**

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="UseVoiceDomainForEmergencyCallBranding"
    Description="Use to let the voice domain decide whether to use 'Emergency calls
only' or 'No service' in the
    phone UI branding. "
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Phone/PhoneSettings">
            <Setting Name="UseVoiceDomainForEmergencyCallBranding" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set the `UseVoiceDomainForEmergencyCallBranding` setting to one of the following values:

VALUE	DESCRIPTION
0 or 'False'	The OS inspects the registration state to decide the emergency call branding. This is the default OS behavior.
1 or 'True'	The voice domain decides the emergency call branding.

If `UseVoiceDomainForEmergencyCallBranding` is set to 1, the phone will not display **Emergency calls only** in the following cases. Instead, it will display **No service**.

- If the system type [RILSYSTEMTYPE] is NONE, which means there is no signal.
- If the system type is LTE but there is no voice domain. This situation can occur in these cases:
  - In LTE networks being used by a non-VoLTE capable device without 3G coverage.
  - In forbidden LTE networks.

However, if you do not set `UseVoiceDomainForEmergencyCallBranding` to 1, or the setting is missing, the device may display **Emergency calls only** in the above situations.

**Testing steps:**

Work with your mobile operator partner to test this customization on their network.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Visual voicemail

10/2/2018 • 7 minutes to read • [Edit Online](#)

**Visual voicemail** supports both traditional voicemail (retrieved through a phone call) and visual voicemail. Users can select between traditional voicemail and visual voicemail when they first attempt to access voicemail. If the mobile operator does not support this visual voicemail implementation, the user will only see the traditional voicemail option. For mobile operators that have their own particular brand that they want to use instead of visual voicemail, partners can rebrand all instances of **visual voicemail** in the Windows 10 Mobile UI to use the operator's brand.

The mobile operator visual voicemail system must be an OMTP-compliant system that meets the following requirements.

- Uses the AMR-NB codec for incoming voicemail messages.
- Sends all SMS-MT as port-directed SMS.
- Sends all SMS-MT with 7-bit default or UCS2 encoding.
- Supports enabling and disabling the visual voicemail feature on the phone by using ACTIVATE and DEACTIVATE SMS-MO messages.

The visual voicemail implementation on the phone is based on the [OMTP visual voice mail interface specification](#). Visual voicemail support on Windows Phone 8.1 was tested on OMTP-based protocols by Comverse and Alcatel Lucent. Other OMTP-based protocols like Streamwide may also be supported, although tests were performed only on Comverse and Alcatel Lucent. Variations from the OMTP standard may result in unsupported scenarios.

The following table shows the extent of support for the features recommended by OMTP. The features marked "Partially supported" provide a button to enable the user to call in to the voicemail system and change the settings over the phone.

FEATURE RECOMMENDED BY OMTP	SUPPORT IN WINDOWS PHONE
IMAP4 message retrieval	Supported
Local visual voicemail store creation	Supported
Hide visual voicemail store from user	Supported
Display non-audio messages	Not supported
Codec support: AMR 12.2k	Supported
Codec support: WAV g711a	Not supported
Codec support: WAV g711u	Not supported
Codec support: QCELP 13.3k	Not supported

Codec support: EVRC 13.3k	Not supported
Mark incoming visual voicemail messages as \Seen, \Deleted	Supported
Deposit visual voicemail messages	Not supported
Forward visual voicemail messages	Not supported
Set/Change TUI password	Partially supported
Change TUI language	Partially supported
Close New User tutorial	Supported
Query for storage quota status	Not supported
Enable/disable on-demand audio message transcription	Not supported
Store a custom personal greeting	Not supported
Delete a stored custom personal greeting	Not supported
Store a voice signature	Not supported
Enable/disable custom personal greeting	Not supported
Retrieve and store provisioning status and credentials	Supported
Activate and deactivate visual voicemail	Supported

**Constraints:** None

## Instructions

To configure visual voicemail for a mobile operator, the OEM must add setting several settings depending on the OMTP-based protocol being used by the operator.

## NOTE

Visual voicemail settings have already been set for AT&T, T-Mobile USA, and Deutsche Telekom AG (DTAG), and no further configuration is required for these three mobile operators.

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="VisualVoicemail"
    Description="Use to configure visual voicemail settings in the phone image."
    Owner=""
    OwnerType="OEM">

    <Static>
        <Settings Path="Phone/VoicemailRegistrationTable">
            <!-- The MCCMNC macro allows you to set multiple MCCMNC\VVMMMO pairs.
                The Value stored here will be the key for the Table. -->
            <Setting Name="ProviderRegistration/${MCCMNC}" Value="" />
            <Setting Name="ProviderRegistration/${MCCMNC}" Value="" />
        </Settings>
        <!-- The VVMMO is the value stored in the MCCMNC setting. This macro allows you to create multiple
            table entries. -->
        <Settings Path="Phone/VoicemailRegistrationTable/${VVMMO}">
            <Setting Name="CLSIDProvider" Value="" />
            <Setting Name="CLSIDAccessor" Value="" />
            <Setting Name="ProtocolVariant" Value="" />
            <Setting Name="IncomingPort" Value="" />
            <Setting Name="ClientType" Value="" />
            <Setting Name="DeviceType" Value="" />
            <Setting Name="InitialSmsDestinationNumber" Value="" />
            <Setting Name="EncryptedSmsSupported" Value="" />
            <Setting Name="KeyData" Value="" />
            <Setting Name="ImapPortOverride" Value="" />
            <Setting Name="TokenLogin" Value="" />
            <Setting Name="SuppressSsl" Value="" />
            <Setting Name="IgnoreLegacyNotifications" Value="" />
            <Setting Name="Branding" Value="" />
        </Settings>
        <Settings Path="Phone/VoicemailRegistrationTable/${VVMMO}">
            <Setting Name="CLSIDProvider" Value="" />
            <Setting Name="CLSIDAccessor" Value="" />
            <Setting Name="ProtocolVariant" Value="" />
            <Setting Name="IncomingPort" Value="" />
            <Setting Name="ClientType" Value="" />
            <Setting Name="DeviceType" Value="" />
            <Setting Name="InitialSmsDestinationNumber" Value="" />
            <Setting Name="EncryptedSmsSupported" Value="" />
            <Setting Name="KeyData" Value="" />
            <Setting Name="ImapPortOverride" Value="" />
            <Setting Name="TokenLogin" Value="" />
            <Setting Name="SuppressSsl" Value="" />
            <Setting Name="IgnoreLegacyNotifications" Value="" />
            <Setting Name="Branding" Value="" />
        </Settings>
    </Static>
</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set multiple MCCMNC\VVMMO pairs by adding the following entry in your customization answer file.

```
<Settings Path="Phone/VoicemailRegistrationTable">
  <Setting Name="ProviderRegistration/${MCCMNC}" Value="" />
</Settings>
```

- Replace `$(MCCMNC)` with the MCCMNC for the mobile operator. For example, 99999.
- Set the corresponding `Value` to the name of the VVMMO. For example, *Contoso*.
- Add and set as many MCCMNC\VVMMO pairs as you need for each mobile operator ID. For example, if you are adding another VVMMO called Fabrikam with MCC/MNC of 999/10, your entries will look like this:

```
<Settings Path="Phone/VoicemailRegistrationTable">
  <Setting Name="ProviderRegistration/99999" Value="Contoso" />
  <Setting Name="ProviderRegistration/99910" Value="Fabrikam" />
</Settings>
```

- For each mobile operator ID defined in the previous step, you must define the applicable settings for that mobile operator by adding the following settings in your customization answer file.

```
<Settings Path="Phone/VoicemailRegistrationTable/${VVMMO}">
  <Setting Name="CLSIDProvider" Value="" />
  <Setting Name="CLSIDAccessor" Value="" />
  <Setting Name="ProtocolVariant" Value="" />
  <Setting Name="IncomingPort" Value="" />
  <Setting Name="ClientType" Value="" />
  <Setting Name="DeviceType" Value="" />
  <Setting Name="InitialSmsDestinationNumber" Value="" />
  <Setting Name="EncryptedSmsSupported" Value="" />
  <Setting Name="KeyData" Value="" />
  <Setting Name="ImapPortOverride" Value="" />
  <Setting Name="TokenLogin" Value="" />
  <Setting Name="SuppressSsl" Value="" />
  <Setting Name="IgnoreLegacyNotifications" Value="" />
</Settings>
```

- Replace `$(VVMMO)` with the name of the VVMMO. For example, *Contoso*.
- Set only the applicable settings that apply to the VVMMO and are required depending on the OMTP-based protocol being used. Note that you do not have to set all of these if they are not supported. The following table describes the values to use and indicates if the keys are required depending on the OMTP-based protocol being used.

KEY NAME	TYPE	GENERIC OMTP	COMVERSE	ALCATEL LUCENT	DETAILS
----------	------	--------------	----------	----------------	---------

Key Name	Type	Generic OMTP	Converse	Alcatel Lucent	Details
CLSIDProvider	REG_SZ	Required	Required	Required	Use {039B8E0E-EA5E-4801-96CD-71E7B343F03F} for an OMTP visual voicemail server or Comverse visual voicemail server. Use {C9804AB2-60B0-4AFF-8205-E30E591F145B} for an Alcatel Lucent visual voicemail server.
CLSIDAccessor	REG_SZ	Required	Required	Required	Use the value {BC371B86-031F-4BD7-9E7D-FB5DF7D1D8C3}
ProtocolVariant	REG_SZ	Required	Required	--	OMTP protocol version ("pv"). Use "ProtocolVariant" = "omtp" for generic OMTP systems, or "ProtocolVariant" = "comverse" for implementations that use Comverse systems.
IncomingPort	REG_DWORD	Required	Required	Required	SMS-MT application port ("pt")
ClientType	REG_SZ	Required	Required	--	An identifier for the category of devices, which can be set to any string. ("ct")
DeviceType	REG_SZ	Required	Required	--	A second-level Converse-specific device type identifier.
InitialSmsDestinationNumber	REG_SZ	Required	Required	--	Phone number to use for SMS-MO messages for visual voicemail such as ACTIVATE or DEACTIVATE ("dn").

Key Name	Type	Generic OMTP	Converse	Alcatel Lucent	Details
EncryptedSmsSupported	REG_DWORD	Not required	Not required	--	Specifies whether 3DES encrypted SMS is supported. Use a value of 0 to indicate it is not supported. Use 1 to indicate this feature is supported.
KeyData	REG_BINARY	Required if EncryptedSmsSupported is set to 1.	Required if EncryptedSmsSupported is set to 1.	--	The binary key to use for encrypted SMS.
ImapPortOverride	REG_DWORD	--	--	Not required	Specifies the IMAP port to use regardless of the message contents. This feature should be turned on only for mobile operators that require it.
TokenLogin	REG_DWORD	--	--	Not required	Enables the use of token-based login instead of traditional username and password. This feature should be turned on only for mobile operators that require it.
SuppressSsl	REG_DWORD	--	--	Not required	Ignores any directive in the message payload to use SSL and forces non-SSL IMAP. This feature should be turned on only for mobile operators that require it. Use a value of 0 to indicate the feature is off; use 1 to indicate it is turned on.

Key Name	Type	Generic OMTP	Converse	Alcatel Lucent	Details
IgnoreLegacyNotifications	REG_DWORD	Not required	Not required	Not required	<p>Specifies whether legacy voicemail notifications should be ignored when visual voicemail is enabled. If the ignore legacy voicemail notification feature is enabled, legacy message waiting indicator SMS messages are ignored (i.e. these will not trigger a visual voicemail sync). If this feature is absent or not enabled, legacy voicemail MWI messages will cause a visual voicemail sync to be initiated.</p> <p>This feature should be turned on only for mobile operators that require it.</p> <p>This feature is not enabled by default.</p> <p>Use a value of 0 to indicate the feature is off; use 1 to indicate it is turned on.</p>

- For mobile operators that have their own particular brand that they want to use instead of visual voicemail, partners can rebrand all instances of **Visual voicemail** in the Windows device UI to use the operator's brand.

To do this, set the value for `Branding` to the specific name that the mobile operator is using for visual voicemail. For example, you can set the value to *Contoso Voice Inbox*.

#### NOTE

This setting does not support a resource-only DLL for localized strings so you need to set the new string directly as the value.

## Testing

Work with your mobile operator to obtain the settings and values that you need to configure visual voicemail and the value to use for `Branding`.

Once you have configured the visual voicemail settings and the branding, work with the mobile operator to test

this customization on their network and verify that all instances of **Visual voicemail** in the Windows device UI have been replaced with the custom brand that you specified.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Voice over LTE call indication

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can add a string to the phone's call progress screen to indicate if the active call is a voice over LTE (VoLTE) call depending on whether the phone call is in high quality voice status such as when using AMR-WB codec. The high quality voice status is determined by the modem and RIL implementation. This string is combined with the PLMN for the mobile operator and is only shown if the combination of the PLMN and the custom string is less than the maximum width available. For example, "Litware VoLTE" will be shown but "Litware Wireless VoLTE" may be too long and may be truncated.

The OS uses the **PhoneMediaQuality** field in the **PH\_CALL\_INFO** structure to determine whether the phone call is in high quality voice status. In the current modem and RIL implementation, PhoneMediaQuality\_High is used to indicate high quality audio during VoLTE calls. **PhoneMediaQuality** must be equal to PhoneMediaQuality\_High and the **VoLTEAudioQualityString** must be set in order for the OS to display the string in the phone's call progress screen.

## Note

Depending on the current modem and RIL implementation, it is possible that the AMR-WB codec is being used and the phone call is in high quality voice status but not a VoLTE call. The OS does not restrict the use of this string in these cases.

**Constraints:** FirstVariationOnly

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="VoLTEAudioQualityString"
    Description="Use to add a string to the call progress screen to indicate if the
call is a voice over LTE call."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Phone/PhoneSettings">
            <Setting Name="VoLTEAudioQualityString" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an **Owner**.
3. Set the **Value** to the string that you want to display in the call progress screen to indicate that the call is a VoLTE call. This string is combined with the PLMN so if the string is 'VoLTE', the resulting string is '*PLMN\_String* VoLTE'. For example, the string displayed in the call progress screen can be 'Litware VoLTE' if the *PLMN\_String* is 'Litware'.

The **Value** you specify for **VoLTEAudioQualityString** must exceed 10 characters.

## Note

This customization does not support a resource-only DLL for localized strings so you need to set the new string directly as the value.

**Testing steps:**

1. Flash the build containing this customization to a phone.
2. Work with your mobile operator partner to test this customization on their network.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Voicemail number for CDMA phones

10/2/2018 • 2 minutes to read • [Edit Online](#)

CDMA mobile operator partners who do not have the voicemail numbers on the device SIM can configure the voicemail number for their devices.

If the voicemail number is not on the SIM and the registry key is not set, the default voicemail will not be set and the user will need to set the number.

**Constraints:** None

This customization supports: **per-IMSI** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="MOSimFallbackVoicemailNumber"
    Description="Use to configure the voicemail number for CDMA phones with no
voicemail numbers on the device."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Phone/PerSimSettings/$_IMSI/Critical">
            <Setting Name="MOSimFallbackVoicemailNumber" Value="" />
        </Settings>
    </Variant>
</ImageCustomizations>
```

2. Specify an `Owner`.

3. For the **per-IMSI** case:

- a. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
  - b. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
4. Set `MOSimFallbackVoicemailNumber` to the voicemail number that you want to use for the device.

**Testing steps:**

1. Flash a build containing this customization to a device.
2. Go to the **Phone** settings screen.
3. Verify that the voicemail number matches the phone number you specified for `Value`.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Customizations for photos, music, and videos

10/2/2018 • 2 minutes to read • [Edit Online](#)

Contains the customizations you can configure for photos, music, and videos.

## In this section

TOPIC	DESCRIPTION
<a href="#">Add OEM lens apps as options for the default camera</a>	OEMs can add lens apps as options for the default camera.
<a href="#">Audio volume limitation</a>	OEMs can configure a setting to display a visual warning when the volume level of the phone exceeds a certain permitted threshold.
<a href="#">Configure OEM lens apps to launch above the lock screen</a>	OEM can configure lens apps to launch above the lock screen.
<a href="#">Configure the FM radio</a>	The BSP provided by the SoC vendor includes support for the FM radio. OEMs can determine whether to show the FM radio app to users, and configure the FM radio frequency for specific regions.
<a href="#">Maximum enumerable photo size</a>	For phones that have the hardware capability to capture various resolutions, partners can specify the resolution limit for photos that can be accessed by third party apps.
<a href="#">Reset the audio volume limitation and warning</a>	OEMs can set the device to reset the audio volume limit and show the volume level warning every time the volume level exceeds a certain permitted threshold for a certain length of time.
<a href="#">Settings for capture mode, burst support, and burst storage duration</a>	OEMs can configure burst support on the device, the default capture mode, and the default number of days to store the bursts captured on the device.
<a href="#">Video over LTE</a>	Partners can customize specific settings and behavior for Video over LTE to meet mobile operator requirements.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Add OEM lens apps as options for the default camera

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can add lens apps as options for the default camera.

OEMs can add lens apps to be shown in the **Pressing the camera button opens** screen within the camera's settings CPL. We recommend that OEMs install only up to five (5) lens apps. The lens apps can be preloaded or installed from the Microsoft Store. When the lens apps are installed on the phone, the apps become available for the user to set as the default camera.

For more information about writing lens apps, see the Windows SDK documentation.

## **Limitations and restrictions:**

- Partners must not remove or modify any app IDs that Microsoft has configured in the kit. The app IDs added by Microsoft do not count against the number of lens app IDs that partners can add.

**Constraints:** None

## Instructions

- Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="OEMLensApps"
    Description="Use to add lens apps to be shown in the Settings > applications >
photos+camera >
    Pressing the camera button opens screen."
    Owner=""
    OwnerType="OEM">
<Static>

    <!-- Preload an OEM lens app -->
    <Applications>
        <!-- For each app, specify the source (.xap/.appx), license, and ProvXML files. -->
        <Application Source="">
            License=""
            ProvXML="" />
    </Applications>

    <!-- Replace $(LensAppGuid) with the app ID of the lens app you want to show in the camera CPL -->
    <Settings Path="Photos/LensApps/$(LensAppGuid)">
        <!-- Set the value to the friendly name of the OEM lens app -->
        <Setting Name="Title" Value="" />
    </Settings>

    <!-- You can add up to 5 OEM lens apps to show in the camera CPL -->
    <Settings Path="Photos/LensApps/$(LensAppGuid)">
        <Setting Name="Title" Value="" />
    </Settings>

    <Settings Path="Photos/LensApps/$(LensAppGuid)">
        <Setting Name="Title" Value="" />
    </Settings>

    <Settings Path="Photos/LensApps/$(LensAppGuid)">
        <Setting Name="Title" Value="" />
    </Settings>

    <Settings Path="Photos/LensApps/$(LensAppGuid)">
        <Setting Name="Title" Value="" />
    </Settings>

</Static>

</ImageCustomizations>

```

2. Specify an `Owner`.
3. If you are preloading a lens app, add an **Applications** parent element and add an **Application** child element to correspond to each lens app that you are preloading. For each **Application**, specify the `Source` (.xap/.appx), `License`, and `ProvXML` files that correspond to the lens app you are preloading.
4. To specify the OEM lens app to show in the camera CPL:
  - a. Replace `$(LensAppGuid)` with the app ID of the lens app you want to show in the camera CPL.
  - b. Replace `Title` with the friendly name of the lens app. For example, *Contoso Fish Eye Lens*.
  - c. Specify up to 5 lens apps by creating a registry entry for each as shown in the preceding example. For example, to configure two lens apps to show in the camera CPL, you need to add the following registry entries:

```
<Settings Path="Photos/LensApps/{00000000-0000-0000-0000-000000000000}">
    <Setting Name="Title" Value="Contoso Fish Eye Lens" />
</Settings>

<Settings Path="Photos/LensApps/{00000000-0000-0000-1000-000000000000}">
    <Setting Name="Title" Value="Contoso Sepia Lens" />
</Settings>
```

## Testing

1. Flash the build containing this customization to a phone.
2. Install the OEM lens app(s).
3. Go to the **Settings > applications > photos+camera** screen and verify that the lens apps that you have specified show up as one of the choices under **Pressing the camera button opens**.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Audio volume limitation

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can configure a setting to display a visual warning when the volume level of the phone exceeds a certain permitted threshold.

Some regions and markets have a volume limitation requirement, which limits the audio volume levels for portable devices like phones and MP3 players. To comply with this requirement, OEMs can configure a setting to display a visual warning when the volume level of the phone exceeds a certain permitted threshold (for example, 85 dB according to European Audio Standards) when audio is playing through the user's headphones or device speakers. User acknowledgment is required before the volume limit is exceeded.

In addition, the phone will keep track of the amount of time that music and video is played at the permitted threshold and display the warning again if and when the user has been listening above the permitted threshold for at least 20 cumulative hours.

To reset the audio volume limit and show the volume level warning every time the volume level exceeds a certain permitted threshold, see [Reset the audio volume limitation and warning](#).

**Constraints:** None

**Instructions:**

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="VolumeLimit"
    Description="Use to display a visual warning when the volume level exceeds a
    certain permitted threshold."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="VolumeLimit">
            <Setting Name="EnableVolumeLimit" Value="" />
            <Setting Name="VolumeLimit" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set `EnableVolumeLimit` to one of the following values:

VALUE	DESCRIPTION
0 or 'Disabled'	Disables volume limits.
1 or 'Enabled'	Enables volume limits and displays a warning when the specified value for <code>VolumeLimit</code> is reached.

4. Set `VolumeLimit` to a value from 10 to 29 (inclusive). For devices sold in the EU, this value should map to 85 dB volume level. This value is also used as the maximum allowed volume in Kid's Corner for media volume.

**Testing steps:**

1. Flash a build containing this customization to a phone.
2. Steadily increase the volume and verify that a warning appears when the volume level has reached the limit that you have set.
3. After accepting the warning, go to the music hub and select a song to play. After 20 hours of cumulative listening above the permitted volume threshold, verify that the warning is displayed again.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Configure OEM lens apps to launch above the lock screen

10/2/2018 • 4 minutes to read • [Edit Online](#)

OEM can configure lens apps to launch above the lock screen.

For lens apps that OEMs have configured to be user selectable default camera app, OEMs can add functionality to launch these lens apps above the lock screen when the user presses the camera button.

Microsoft recommends that OEMs configure only up to five (5) lens apps capable of running above the lock screen. The lens apps can be preloaded by OEMs or installed by users from the Microsoft Store.

**Constraints:** None

## Instructions:

The steps for configuring an OEM lens app to run above the lock screen are very similar to [Adding OEM lens apps as options for the default camera](#), but with the additional step of designating it as a lens app that can run above the lock screen.

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="OEMLensAboveLock"
    Description="Use to configure an OEM lens app to launch above the lock screen when
the user presses the camera button."
    Owner=""
    OwnerType="OEM">

    <Static>

        <!-- Preload an OEM lens app -->
        <Applications>
            <!-- For each app, specify the source (.xap/.appx), license, and ProvXML files. -->
            <Application Source=""
                License=""
                ProvXML="" />
        </Applications>

        <!-- Replace $(LensAppGuid) with the app ID of the lens app you want to show in the camera CPL -->
        <Settings Path="Photos/LensApps/$(LensAppGuid)">

            <!-- Set the value to the friendly name of the OEM lens app -->
            <Setting Name="Title" Value="" />

            <!-- Set this to the version of the OEM lens app version that you want to launch above the lock
screen -->
            <Setting Name="MinVersionAboveLock" Value="" />
        </Settings>

        <!-- You can add up to 5 OEM lens apps to show in the camera CPL. For other OEM lens apps that you
want to enable to run
            above the lock screen, you must set the MinVersionAboveLock setting for each of these. -->
    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.
3. If you are preloading a lens app, add an **Applications** parent element and add an **Application** child element to correspond to each lens app that you are preloading. For each **Application**, specify the `Source` (.xap/.appx), `License`, and `ProvXML` files that correspond to the lens app you are preloading.

#### 4. To specify the OEM lens app to show in the camera CPL:

- a. Replace `$(LensAppGuid)` with the app ID of the lens app you want to show in the camera CPL.
- b. Replace `Title` with the friendly name of the lens app. For example, *Contoso Fish Eye Lens*.
- c. Specify up to 5 lens apps by creating a registry entry for each as shown in the preceding example. For example, to configure two lens apps to show in the camera CPL, you need to add the following registry entries:

```
<Settings Path="Photos/LensApps/{00000000-0000-0000-0000-000000000000}">
  <Setting Name="Title" Value="Contoso Fish Eye Lens" />
</Settings>

<Settings Path="Photos/LensApps/{00000000-0000-0000-1000-000000000000}">
  <Setting Name="Title" Value="Contoso Sepia Lens" />
</Settings>
```

#### 5. To designate an OEM lens app to run above the lock screen:

- a. Add the `MinVersionAboveLock` setting within the settings group for the OEM lens app.
- b. Set the value of `MinVersionAboveLock` to a string that is equal to the version of the first published OEM lens app version that fully complies with the guidelines and requirements outlined in the previous section. If the lens app has an equal or higher version to the value that you set for `MinVersionAboveLock`, the lens app will launch above the lock screen when the camera button is pressed on a PIN-locked screen. Otherwise, the PIN unlock screen shows when the camera button is pressed, and if the user enters the correct password, the lens app will launch.

OEMs may set the value for `MinVerAboveLock` to a sufficiently large version string so that you may release the phone first and later publish an updated lens app to the Windows Phone Store that fully complies with the requirements and guidelines for OEM lens apps that launch above the lock screen.

In the following example, the Contoso Sepia Lens app has been designated as the OEM lens app to run above the lock screen.

```
<Settings Path="Photos/LensApps/{00000000-0000-0000-0000-000000000000}">
  <Setting Name="Title" Value="Contoso Fish Eye Lens" />
</Settings>

<Settings Path="Photos/LensApps/{00000000-0000-0000-1000-000000000000}">
  <Setting Name="Title" Value="Contoso Sepia Lens" />
  <Setting Name="MinVersionAboveLock" Value="1.0.0.0" />
</Settings>
```

#### Testing:

1. Flash the build that contains this customization to a phone.
2. Install the OEM lens app(s).
3. Go to the **Settings > applications > photos+camera** screen and verify that the lens apps that you have specified show up as one of the choices under **Pressing the camera button opens**.

Choose a lens app that you have configured to run above the lock screen and set it as the default camera.

4. Go to **Settings** > **lock screen**, turn **Password** on, then set a password or PIN.
5. Lock the phone.
6. Press and hold the camera button while the phone is locked. Verify that the lens app that you chose in Step 3 is launched.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Configure the FM Radio

10/2/2018 • 2 minutes to read • [Edit Online](#)

The BSP provided by the SoC vendor includes support for the FM radio.

Alternative FM radio components that meet the requirements described in the section [2.7: Wireless communications](#) of the Chassis Requirements Specification can be used. If alternative components are used, the FM miniport driver can be replaced.

## NOTE

If the phone includes an FM radio, it must run the Microsoft-supplied software driver stack, including the port driver.

There are two customization options associated with the FM radio.

- Show the FM radio app to users
- Configure the FM radio frequency for specific regions

Application programming interfaces are not provided for the FM radio.

## Show the FM radio

For devices that include an FM radio chip, OEMs can show **FM Radio** in the Apps list. In addition, OEMs can also set the default [FM radio frequency band](#).

By default, the Windows 10 Mobile FM radio UI is hidden.

**Constraints:** None

### Instructions

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="ShowFMRadioUI"
    Description="Use to show the FM radio UI for devices that include an FM radio
chip."
    Owner=""
    OwnerType="OEM">
    <Static>
        <Settings Path="FmRadio">
            <Setting Name="NotPresent" Value="0" />
        </Settings>
    </Static>
</ImageCustomizations>
```

2. Specify an `Owner`.
3. Set `NotPresent` to 0 to show the **FM Radio** app. > `[!Note]` > Setting `NotPresent` to 1 is not necessary because the radio UI is hidden by default.

### Testing

1. Flash the build containing this customization to a device.

- Verify that **FM radio** is now visible in the Apps list.

## Configure the FM radio frequency band

OEMs can change the default setting for the FM radio receiver to use an appropriate frequency for the market in which the device will be sold.

### NOTE

`NotPresent` must be set to 0 to show the **radio** option in the UI.

### Limitations and restrictions:

- Additional frequency bands cannot be added to the device.
- The user can change the default setting.

**Constraints:** None

### Instructions

- Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="FMRadioRegion"
    Description="Use to change the default frequency band for the FM radio receiver."
    Owner=""
    OwnerType="OEM">
    <Static>
        <Settings Path="FmRadio">
            <Setting Name="NotPresent" Value="0" />
            <Setting Name="RadioRegion" Value="" />
        </Settings>
    </Static>
</ImageCustomizations>
```

- Specify an `Owner`.

- Set `RadioRegion` to specify the default region for the frequency band for the device's FM radio. Set this to one of the following values:

VALUE	DESCRIPTION
1	North America
2	World
3	Japan

### Testing

- Flash the build containing this customization to a device.
- Open the **radio** app.
- In the radio application, verify that the selected region matches the one you specified in `RadioRegion`. To do this, show the context menu by tapping and holding anywhere on the radio screen. In the context menu, tap **settings** to show the settings page.

4. In the **Settings** page, verify that **Region** is set to the default FM radio region that you selected.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Maximum enumerable photo size

10/2/2018 • 2 minutes to read • [Edit Online](#)

For phones that have the hardware capability to capture various resolutions, partners can specify the resolution limit for photos that can be accessed by third party apps.

Only OEM applications have access to the maximum resolution limit.

## Note

This customization is only used by the **Windows.Phone.Media.Capture** service, which is provided in Windows Phone 8.1 for backwards compatibility only. **Windows.Media.Capture** does not support this customization.

**Constraints:** None

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="MaximumEnumerablePhotoSize"
    Description="On phones that can capture multiple resolutions, use to specify the
resolution limit for photos
                                that can be accessed by third party apps."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Camera">
            <Setting Name="MaximumEnumerablePhotoSize" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.
3. Set the value for `MaximumEnumerablePhotoSize` to the photo resolution in pixels. For example, to set 5 megapixels as the maximum enumerable photo size, set `Value` to 5242880 (decimal) or 0x500000 (hexadecimal).

## Testing:

1. Flash the build containing this customization to a phone that has the hardware capability you are testing.
2. Run a test using the `PhotoCaptureDevice.GetAvailableCaptureResolutions` method and check if resolutions higher than the value you specified for `MaximumEnumerablePhotoSize` have been filtered out.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Reset the audio volume limitation and warning

10/2/2018 • 3 minutes to read • [Edit Online](#)

OEMs can set the device to reset the audio volume limit and show the volume level warning every time the volume level exceeds a certain permitted threshold for a certain length of time.

To reset the audio volume limit and show the volume level warning every time the volume level exceeds a certain permitted threshold for at least 20 cumulative hours, OEMs can set the **VolumeThresholdPlayTimeLimit** registry value. By default, the cumulative time limit for the audio volume limit is set to 20 hours and the OS timer that tracks the cumulative time limit is fired every 6 minutes so OEMs must set

**VolumeThresholdPlayTimeLimit** to a value ≤ 19 hours and 54 minutes. When set, the volume drops back to the permitted threshold set in the [Audio volume limitation](#) customization if the user has been listening to music at more than the permitted volume limit and the cumulative listening time is between 19 hours, 54 minutes and 20 hours. The volume limit warning will also reappear.

**Constraints:** None

## Instructions:

1. Create the MCSF policy setting that corresponds to the following registry key:

```
$\HKLM\Microsoft\ZMediaQ\VolumeLimit\VolumeThresholdPlayTimeLimit  
Type: REG_DWORD  
Value: 000117D8
```

### NOTE

Set **Value** to a hexadecimal value that corresponds to 19 hours and 54 minutes or less. In the above example, the value 0x000117D8 corresponds to 71640 seconds or 19 hours and 54 minutes.

For more information about MCSF, see [Managed Centralized Settings Framework \(MCSF\)](#). The following code example shows an MCSF policy setting for the **VolumeThresholdPlayTimeLimit** registry value.

```
<SettingsGroup Path="VolumeLimit">  
    <Setting Name="VolumeThresholdPlayTimeLimit" Description="Resets the audio volume limit and  
    shows the volume level warning  
        every time the volume exceeds the VolumeLimit threshold for at least 20  
        cumulative hours.">  
        <RegistrySource Type="REG_DWORD"  
        Path="HKEY_LOCAL_MACHINE\Software\Microsoft\ZMediaQ\VolumeLimit\VolumeThresholdPlayTimeLimit" />  
    </Setting>  
</SettingsGroup>
```

2. Add the policy setting in the previous step to a .pkg.xml file. The following code example shows the MCSF policy setting within the .pkg.xml file.

```

<?xml version="1.0" encoding="utf-8"?>
<Package xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  Owner=""
  Component=""
  SubComponent=""
  OwnerType="OEM"
  ReleaseType="">
  <Components>
    <SettingsGroup Path="VolumeLimit">
      <Setting Name="VolumeThresholdPlayTimeLimit" Description="Resets the audio volume limit and
shows the volume level warning
every time the volume exceeds the VolumeLimit threshold for at least 20
cumulative hours.">
        <RegistrySource Type="REG_DWORD"
Path="HKEY_LOCAL_MACHINE\Software\Microsoft\ZMediaQ\VolumeLimit\VolumeThresholdPlayTimeLimit" />
      </Setting>
    </SettingsGroup>
  </Components>
</Package>

```

In this example, provide values for the **Owner**, **Component**, **SubComponent**, and **ReleaseType** attributes.

3. Use the .pkg.xml file that contains your MCSF policy setting to generate a package (or .spkg file) that you can add to your OS image.
4. After you've created the .spkg, define the specific types of image builds that you want to contain the package.

For example, the following code snippet shows a sample OEM feature manifest (FM) file that may contain the .spkg that includes the customization:

```

<?xml version="1.0" encoding="utf-8"?>
<FeatureManifest xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate">
  <!-- Sample FM File -->
  <Features>
    <OEM>
      <PackageFile Path="SourceDirectory"
Name="OEMName.SoundCustomizations.VolumeThresholdPlayTimeLimit.spkg">
        <FeatureIDs>
          <FeatureID>VOLUME_THRESHOLD_PLAY_TIME_LIMIT</FeatureID>
        </FeatureIDs>
      </PackageFile>
    </OEM>
  </Features>
</FeatureManifest>

```

In this example, replace *SourceDirectory* with the location that contains the .spkg that you created in Step 3. Also, replace the example *OEMName.SoundCustomizations.VolumeThresholdPlayTimeLimit.spkg* with the name of the .spkg file.

5. Once you've defined the feature, modify your OEMInput.xml file to add a **Features** element (if one doesn't already exist), add a new **OEM** child element (if one doesn't already exist), and add a new **Feature** entry with the name of the feature that you just defined.

For example, the OEMInput.xml entry for the example VOLUME\_THRESHOLD\_PLAY\_TIME\_LIMIT feature may look like the following:

```
<Features>
  <OEM>
    <Feature>VOLUME_THRESHOLD_PLAY_TIME_LIMIT</Feature>
  </OEM>
</Features>
```

For more information, see [OEMInput file contents](#).

6. Build the OS image. For more information, see *Using ImgGen.cmd to generate an image* in [Building a mobile image using ImgGen.cmd](#).

**Testing:**

1. Flash the build that contains this customization to a device.
2. Set up your device and load music on the device.
3. Use the device and headset to play and listen to music. Turn up the volume above the limit that you set for the **VolumeLimit** setting in [Audio volume limitation](#).

Continue to listen to music for 20 or more hours.

4. If you set **VolumeThresholdPlayTimeLimit** to a hexadecimal value that corresponds to 19 hours and 54 minutes (or less), verify that the volume level is reset and the volume warning is shown before you reach 20 cumulative hours of playing music with the volume set above the **VolumeLimit**.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Settings for capture mode, burst support, and burst storage duration

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can configure burst support on the device, the default capture mode, and the default number of days to store the bursts captured on the device.

**Constraints:** ImageTimeOnly

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="BurstAndCaptureModeSettings"
    Description="Use to configure the default capture mode on the device and configure
the settings related to burst."
    Owner=""
    OwnerType="OEM">

    <Static>
```

```
<Settings Path="Photos/OEM">
    <Setting Name="BurstSupported" Value="" />
    <Setting Name="CaptureMode" Value="" />
    <Setting Name="DefaultBurstStorageDuration" Value="" />
</Settings>

</Static>

</ImageCustomizations>
```

```

1. Specify an `Owner`.

2. **To configure burst support on the device**, set `BurstSupported` to one of the following values:

| VALUE                       | DESCRIPTION                                                     |
|-----------------------------|-----------------------------------------------------------------|
| 1 or 'True, supported'      | Burst is supported.<br>This is the default value set by the OS. |
| 0 or 'False, not supported' | Burst is not supported.                                         |

3. **To configure the default capture mode on the device**, set `CaptureMode` to one of the following values:

| VALUE              | DESCRIPTION                                                     |
|--------------------|-----------------------------------------------------------------|
| 1 or 'Burst'       | Burst capture mode.<br>This is the default value set by the OS. |
| 0 or 'Single shot' | Single shot capture mode.                                       |

4. **To configure the default number of days to store the bursts captured on the device**, set

`DefaultBurstStorageDuration` to the number of days you want to keep the bursts on the device. For example, a value of 1 means the bursts will be kept for 1 day.

Microsoft recommends using any of these values: 1, 3, 7 (the default value set by the OS), or 14. A value of 0 indicates that bursts will be stored forever.

**Testing steps:**

1. Flash a build containing this customization to a phone.
2. Launch the camera app.
3. To verify the support for burst mode, check if the burst icon is displayed on the right of the settings bar when the camera app is launched.
4. To verify the default capture mode, check whether single shot or burst mode is selected the first time the camera app was launched. The default should match the value that you specified. For example, if you selected single shot, the camera icon on the right should be twice as big as the other icons.
5. To verify the default burst storage duration, go to the camera **Settings** screen, choose **photo settings...**, and verify if the default value specified in the **Keep unsaved burst photos for** field matches the value that you specified.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Video over LTE

10/2/2018 • 6 minutes to read • [Edit Online](#)

Partners can customize specific settings and behavior for Video over LTE to meet mobile operator requirements.

These include:

- Showing or hiding the LTE video calling switch
- Setting the default value for the switch
- Customizing the name/label of the switch and the description
- Specifying the timeout, in milliseconds, for the device to remain in video transition state
- Enabling video conferencing
- Specifying the amount of time before a video call is downgraded to a voice call due to low video quality
- Hiding the video charges dialog that is displayed when the user turns on the LTE video calling switch

**Constraints:** Some None, some FirstVariationOnly

This customization supports: **per-IMSI** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="VideoOverLTESettings"
    Description="Use to customize the settings for Video over LTE."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>

        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>

        <!-- Add the resource-only dll file and language MUI packages if you are using Windows Phone 8.1
        GDR1 and later versions of the OS -->
        <Settings Path="Localization/MUI">
            <!-- Use to add your base MUI DLL file -->
            <Asset Name="BaseDll" Source="" />

            <!-- Use to specify the language MUI packages (*.dll.mui) for the languages you are supporting and -->
    </Settings>
</Static>
</ImageCustomizations>
```

```

have localized strings for -->
    <Asset Name="LanguageDll/${langid}" Source="" />
    <Asset Name="LanguageDll/${langid}" Source="" />
    <Asset Name="LanguageDll/${langid}" Source="" />
    <!-- Add as many as you need -->
</Settings>
</Static>

<!-- Specify the Variant -->
<Variant Name="">
    <TargetRefs>
        <TargetRef Id="" />
    </TargetRefs>

    <Settings Path="Phone/PerSimSettings/${__IMSI}">
        <Setting Name="ShowVideoCallingSwitch" Value="" />
        <Setting Name="DefaultEnableVideoCalling" Value="" />
        <Setting Name="DefaultEnableVideoCapability" Value="" />
        <Setting Name="ShowVideoCapabilitySwitch" Value="" />
        <Setting Name="AllowVideoConferencing" Value="" />
        <Setting Name="SupressVideoCallingChargesDialog" Value="" />
    </Settings>

    <Settings Path="Phone/PhoneSettings">
        <!-- Note that these settings are FirstVariationOnly -->

        <Setting Name="VideoTransitionTimeout" Value="" />
        <Setting Name="VideoCallingLabel" Value="" />
        <Setting Name="VideoCallingDescription" Value="" />
        <Setting Name="LowVideoQualityTimeout" Value="" />

        <!-- If you enable video conferencing, you can also specify the number of participants that can be
        added to the conference call -->
        <Setting Name="ConferenceCallMaximumPartyCount" Value="" />
    </Settings>
</Variant>
</ImageCustomizations>

```

2. Specify an `Owner`.

3. Add the resource-only .dll file and the language MUI packages (\*.dll.mui) for the languages you are supporting. To do this, follow these steps:

- Add the resource-only .dll that contains the custom display string by setting the `BaseDll` asset to point to the location of your base MUI DLL file. For example: `C:\Path\DisplayStrings.dll`.
- Add the language MUI packages (\*.dll.mui) for all the languages you are supporting and have localized strings for. To do this:
  - Set the asset's `Name` to `LanguageDll/ ${langid}` where `$(langid)` corresponds to the language. For example: `LanguageDll/en-US`.
  - Set the asset's `Source` to the location of the .dll.mui file for that language. For example: `C:\Path\en-us\DisplayStrings.dll.mui`.
  - Repeat the previous steps for the other languages.

The following example shows the customization answer file entries for en-US, fr-CA, and es-MX languages:

```

<Asset Name="LanguageD11/en-US" Source="C:\Path\en-us\DisplayStrings.dll.mui" />
<Asset Name="LanguageD11/fr-CA" Source="C:\Path\fr-CA\DisplayStrings.dll.mui" />
<Asset Name="LanguageD11/es-MX" Source="C:\Path\es-MX\DisplayStrings.dll.mui" />

```

For more information, see [Create a resource-only .dll for localized strings](#).

4. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
5. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
6. To show or hide the LTE video calling switch, set `ShowVideoCallingSwitch` to one of the following values:

VALUE	DESCRIPTION
0 or 'False'	Hides the LTE video calling switch. This is the default OS value.
1 or 'True'	Shows the LTE video calling switch.

**\*\*Note\*\***

This setting does not set the default switch value. To do that, set `DefaultEnableVideoCalling`.

1. To set the initial default value for the LTE video calling switch, set `DefaultEnableVideoCalling` to one of the following values:

VALUE	DESCRIPTION
0 or 'False'	Sets the LTE video calling switch to Off. This is the default OS value.
1 or 'True'	Sets the LTE video calling switch to On.

2. To set the initial value for LTE video capability sharing, set `DefaultEnableVideoCapability` to one of the following values:

VALUE	DESCRIPTION
0 or 'False'	Sets the LTE video capability sharing to Off. This is the default OS value.
1 or 'True'	Sets the LTE video capability sharing to On.

3. To specify whether to show the video capability sharing switch on the phone **Settings** screen, set `ShowVideoCapabilitySwitch` to one of the following values:

VALUE	DESCRIPTION
0 or 'False'	Hides the video capability sharing switch. This is the default OS value.
1 or 'True'	Shows the video capability sharing switch.

4. To enable LTE video calls to be merged into a conference call, set `AllowVideoConferencing` to one of the following values:

VALUE	DESCRIPTION
0 or 'False'	Disables LTE video calls from being merged into a conference call. This is the default OS value.
1 or 'True'	Enables LTE video calls to be merged into a conference call.  If you are enabling LTE video calls, you can also specify the maximum number of participants or callers that can be added to the video conference by setting <code>ConferenceCallMaximumPartyCount</code> . For more information, see <a href="#">Maximum number of participants in a VoLTE conference call</a> .

If this setting is not enabled, the conference option in the UI will never be available. It will always show up as disabled (greyed out).

**\*\*Note\*\***

Video conference support is dependent on support by the mobile operator and the device chipset.

1. To show or hide the video charges dialog that is displayed when the user turns on the LTE video calling switch, set `SupressVideoCallingChargesDialog` to one of the following values:

VALUE	DESCRIPTION
0 or 'False'	Shows the video calling charges dialog. This is the default OS value.
1 or 'True'	Hides the video calling charges dialog.

2. To set the time, in milliseconds, to wait for the response to the request to transition a VoLTE call to video, set `VideoTransitionTimeout`. You can set the value to a number between 10000 and 30000, inclusive. If you set the value to 0, the OS uses the default value of 30000 (30 seconds).

An alert tone is played multiple times during this request. Each alert tone will be played in 10 second increments. The number of alert tones is determined by the request time out value divided by 10. For example:

- If you set the value to 30000 (or 30 seconds), the alert tone will play three times at 0, 10 and 20 seconds, and then 10 seconds later the request will time out.
  - If you set the value to 25000 (or 25 seconds), the alert tone will play three times at 0, 10, and 20 seconds, then 5 seconds later the request will time out.
3. To customize the name or label of the LTE video calling switch and the description for the switch, set the value for the following settings:
- a. To customize the switch name or label, set the `VideoCallingLabel` value to the name of the resource-only .dll file and specify the string offset. For example: `@DisplayStrings.dll,-101`.  
Replace `DisplayStrings.dll` with the name of your .dll file and replace `Offset` with the correct offset for the localized string.
  - b. To customize the switch description, set the `VideoCallingDescription` value to the name of the resource-only .dll file and specify the string offset. For example: `@DisplayStrings.dll,-102`.  
Replace `DisplayStrings.dll` with the name of your .dll file and replace `Offset` with the correct offset for the localized string.

4. To set the timer, in milliseconds, to automatically drop video support from an active video call when the video calling quality is low, set `LowVideoQualityTimeout`. This will transition the call to a VoLTE call. Set the value to a number between 0 and 120000, inclusive.

A value of 0 disables the timer. This is also the OS default value.

#### Note

Support for this feature is dependent on support by the mobile operator and the chipset.

#### Testing:

1. Flash the build containing this customization to a device.
2. Go to the **Phone** settings screen.
3. Verify whether the switches are visible or the correct settings and values are showing up based on the values you specified for the Video over LTE settings.
4. If you customized the switch label and description, verify that the correct localized strings are showing up based on the device language.
5. If the right conditions are met, verify that the timeouts work as expected.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Customizations for ringtones and sounds

10/2/2018 • 2 minutes to read • [Edit Online](#)

Describes the customizations that you can configure for ringtones and sounds played by the device.

## In this section

TOPIC	DESCRIPTION
<a href="#">Additional alarms</a>	Partners can add one additional alarm sound to the phone for use in the Alarm & Clocks app. Partners can also set a new default alarm.
<a href="#">Additional notification sounds</a>	Partners can add up to three new notification sounds and a reminder sound. In addition, you can also specify a default notification sound for messaging, voicemails, and reminders.
<a href="#">Additional ringtones</a>	OEMs and mobile operators can each preload a set of custom ringtone files on Windows mobile devices, and they can set a default ringtone.
<a href="#">Call drop and call waiting sounds</a>	OEMs can customize the call drop and call waiting sounds.
<a href="#">Camera shutter sound</a>	The camera shutter sound that occurs when the user takes a picture or starts filming a video can be turned off by removing the Camera shutter option from the Sounds settings screen.
<a href="#">Ringtone store application</a>	Partner apps can be used to sell ringtones to users. The app owner must provide the service for the ringtone catalog and to manage downloads. Users are shown an option to Get more ringtones in the ringtone picker, from which they can automatically launch the ringtone store application.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Additional alarms

10/2/2018 • 3 minutes to read • [Edit Online](#)

Partners can add one additional alarm sound to the phone for use in the **Alarm & Clocks** app. Partners can also set a new default alarm.

Alarms should be a maximum of 100 KB and have a length of 5 to 15 seconds. They must be in .wma format, with a compression of 128 kbit/s for stereo or 64 kbit/s mono. Partner sounds should play at an appropriate volume relative to other sounds and ringtones, and there should be minimal distortion from the device speaker, at full volume.

## **Limitations and restrictions:**

- The names of the sounds must be localized for all display languages that ship on the device.
- Sound files must be approved by Microsoft.
- Partners must not move, delete, or modify the alarms provided by Microsoft.

Partners must keep the following design considerations in mind when implementing additional alarms:

**File size:** Ringtone recommended maximum 150 KB; Alarm 100 KB (others as small as possible)

**Format:** .wma

**Compression:** WMA (128 kbps/stereo; 64 kbps/mono)

**Sound length:** Ringtones 5-15 seconds; Alarm 5-15 seconds; Calendar 1-3 seconds; Alerts 0.5-1.5 seconds

**Appropriate volumes:** Sounds should be appropriately balanced with ringtones and system sounds that ships as part of the OS.

**Minimal distortion** from phone speaker, at full volume.

**Constraints:** FirstVariationOnly

## **Instructions:**

### **To add an additional alarm sound**

1. Create a .dll that contains the alarm display name. For more information on how to do this, see [Create a resource-only .dll for localized strings](#).
2. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="AdditionalAlarms"
    Description="Use to add additional alarm sounds and set a new default alarm sound."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Localization/MUI">
            <!-- Use to add your base MUI DLL file -->
            <Asset Name="BaseDll" Source="" />

            <!-- Use to specify the language MUI packages (*.dll.mui) for the languages you are supporting and have localized strings for -->
            <Asset Name="LanguageDll/${langid}" Source="" />
            <Asset Name="LanguageDll/${langid}" Source="" />
            <Asset Name="LanguageDll/${langid}" Source="" />
            <!-- Add as many as you need -->
        </Settings>

        <!-- Use to add one additional alarm sound -->
        <Settings Path="EventSounds">
            <Asset Name="AlarmSounds" DisplayName="@DisplayStrings.dll,-Offset" Source="" />
        </Settings>

    </Static>

</ImageCustomizations>

```

3. Specify an `Owner`.

4. Add the resource-only .dll that contains the alarm display name by setting the `BaseDll` asset to point to the location of your base MUI DLL file. For example: `C:\Path\DisplayStrings.dll`.
5. Add the language MUI packages (\*.dll.mui) for all the languages you are supporting and have localized strings for. To do this:

- Set the asset's `Name` to `LanguageDll/ ${langid}` where `$(langid)` corresponds to the language. For example: `LanguageDll/en-US`.
- Set the asset's `Source` to the location of the .dll.mui file for that language. For example: `C:\Path\en-us\DisplayStrings.dll.mui`.
- Repeat the previous steps for the other languages.

The following example shows the customization answer file entries for en-US, fr-CA, and es-MX languages:

```

<Asset Name="LanguageDll/en-US" Source="C:\Path\en-us\DisplayStrings.dll.mui" />
<Asset Name="LanguageDll/fr-CA" Source="C:\Path\fr-CA\DisplayStrings.dll.mui" />
<Asset Name="LanguageDll/es-MX" Source="C:\Path\es-MX\DisplayStrings.dll.mui" />

```

6. Add one additional alarm sound by adding an `AlarmSounds` asset. To do this:

- Set the asset's `Name` to `AlarmSounds`.
- Set the `DisplayName` to the name of the resource-only .dll file and specify the string offset. Replace `DisplayStrings.dll` with the name of your .dll file and replace `Offset` with the correct offset for the localized string. For example: `@DisplayStrings.dll,-104`

- Set **Source** to the full path to the custom alarm sound on your development machine. For example: `C:\Path\NewAlarmSound.wma`.

If you are setting the default alarm sound in addition to adding other alarm sound files, see the *To set a new default alarm* section.

### To set a new default alarm

1. Create a customization answer file using the contents shown in the following example:

```
<!-- Use to set a new default alarm sound -->
<Settings Path="EventSounds">
    <Setting Name="DefaultAlarmSound" Value="" />
</Settings>
```

2. Set the **Value** of the default alarm sound to the file name of the default alarm sound you want to use. For example: `NewAlarmSound.wma`

### Testing:

1. Flash the build containing this customization and multiple display languages to a mobile device.
2. Go to **Alarms & Clock** in the apps list.
3. Tap the + or add button to create a new alarm.
4. Choose **Pick from ringtones** and verify the custom alarm sound is in the **Sounds** list.
5. If a new default alarm sound is set, verify the **Sound** drop-down list defaults to the specified alarm sound.
6. Verify the custom alarm name is correct for all display languages on the device.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Additional notification sounds

10/2/2018 • 4 minutes to read • [Edit Online](#)

Partners can add up to three new notification sounds and a reminder sound. In addition, you can also specify a default notification sound for messaging, voicemails, and reminders.

Alerts should have a length of 1 to 3 seconds and be as small as possible in size. They must be in .wma format, with a compression of 128 kbit/s for stereo or 64 kbit/s for mono. Partner sounds should play at an appropriate volume relative to other sounds and ringtones, and there should be minimal distortion from the device speaker, at full volume.

## **Limitations and restrictions:**

- The names of the sounds must be localized for all display languages that ship on the device.
- Sound files must be approved by Microsoft.
- Partners must not move, delete, or modify the notification sounds provided by Microsoft.

Partners must keep the following design considerations in mind when implementing additional notification sounds:

**File size:** Ringtone recommended maximum 150 KB; Alarm 100 KB(others as small as possible)

**Format:** .wma

**Compression:** WMA (128 kbps/stereo; 64 kbps/mono)

**Sound length:** Ringtones 5-15 seconds; Alarm 5-15 seconds; Calendar 1-3 seconds; Alerts 0.5-1.5 seconds

**Appropriate volumes:** Sounds should be appropriately balanced with the ringtones and system sounds that's part of the OS.

**Minimal distortion** from phone speaker, at full volume.

**Constraints:** FirstVariationOnly

## **Instructions:**

### **To add additional notification sounds**

1. Create a .dll that contains the notification sound display name. For more information on how to do this, see [Create a resource-only .dll for localized strings](#).
2. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="AdditionalNotifications"
    Description="Use to add additional notification sounds and set new default
notification sounds for
                                messaging, voicemail, or calendar reminders."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Localization/MUI">
            <!-- Use to add your base MUI DLL file -->
            <Asset Name="BaseDll" Source="" />

            <!-- Use to specify the language MUI packages (*.dll.mui) for the languages you are supporting and
have localized strings for -->
            <Asset Name="LanguageDll/${langid}" Source="" />
            <Asset Name="LanguageDll/${langid}" Source="" />
            <Asset Name="LanguageDll/${langid}" Source="" />
            <!-- Add as many as you need -->
        </Settings>

        <Settings Path="EventSounds">
            <!-- Use to add up to three new notification sounds and one additional notification sound for
calendar reminders -->
            <Asset Name="NotificationSounds" DisplayName="@DisplayStrings.dll,-Offset" Source="" />
            <Asset Name="NotificationSounds" DisplayName="@DisplayStrings.dll,-Offset" Source="" />
            <Asset Name="NotificationSounds" DisplayName="@DisplayStrings.dll,-Offset" Source="" />
            <Asset Name="NotificationSounds" DisplayName="@DisplayStrings.dll,-Offset" Source="" />
        </Settings>

    </Static>

</ImageCustomizations>

```

3. Specify an `Owner`.

4. Add the resource-only .dll that contains the notification sounds' display names by setting the `BaseDll` asset to point to the location of your base MUI DLL file. For example: `C:\Path\DisplayStrings.dll`.
5. Add the language MUI packages (\*.dll.mui) for all the languages you are supporting and have localized strings for. To do this:
  - Set the asset's `Name` to `LanguageDll/ ${langid}` where `$(langid)` corresponds to the language. For example: `LanguageDll/en-US`.
  - Set the asset's `Source` to the location of the .dll.mui file for that language. For example: `C:\Path\en-us\DisplayStrings.dll.mui`.
  - Repeat the previous steps for the other languages.

The following example shows the customization answer file entries for en-US, fr-CA, and es-MX languages:

```

<Asset Name="LanguageDll/en-US" Source="C:\Path\en-us\DisplayStrings.dll.mui" />
<Asset Name="LanguageDll/fr-CA" Source="C:\Path\fr-CA\DisplayStrings.dll.mui" />
<Asset Name="LanguageDll/es-MX" Source="C:\Path\es-MX\DisplayStrings.dll.mui" />

```

6. Add additional notification sounds by adding a `NotificationSounds` asset. To do this:

- Set the asset's `Name` to `NotificationSounds`.

- Set the `DisplayName` to the name of the resource-only .dll file and specify the string offset. Replace `DisplayStrings.dll` with the name of your .dll file and replace `Offset` with the correct offset for the localized string. For example: `@DisplayStrings.dll,-104`.
- Set `Source` to the full path to the custom notification sound on your development machine. For example: `C:\Path\NewVoicemailNotification.wma`.
- Repeat the previous steps for any additional notification sounds. Partners can add up to three new notification sounds and one additional notification sound for reminders.

If you are setting the default notification sound in addition to adding other notification sound files, see the *To set a new sound for messaging, voicemail, or reminders* section.

### To set a new default notification sound for messaging, voicemail, or reminders

- Create a customization answer file using the contents shown in the following code example:

```
<Settings Path="EventSounds">
    <!-- Use to set a new default voicemail notification sound -->
    <Setting Name="DefaultVoicemailAlertSound" Value="" />

    <!-- Use to set a new default reminder sound -->
    <Setting Name="DefaultReminderAlertSound" Value="" />

    <!-- Use to set a new default messaging notification sound -->
    <Setting Name="DefaultMessagingSound" Value="" />

</Settings>
```

- For the default notification sound that you want to configure, set the `Value` to the desired default notification sound file name.

For example, if you only want to set `DefaultMessagingSound` to the messaging sound that you added, set the default messaging sound value to `MessagingSound.wma`.

### Testing:

- Flash the build containing this customization and multiple display languages to a mobile device.
- Go to the **Sounds** settings screen. Verify that all added custom notification sounds appear as expected.
- Go to the **Notifications & actions** screen in **Settings**. From the list, select an email account, **Messaging**, or **Phone**, and then select the **Notification sound** drop-down list to verify all added custom notification sounds for voicemail and messaging appear in the drop-down list.
- Verify the custom notification sound names are correct for all display languages on the device.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Additional ringtones

10/2/2018 • 3 minutes to read • [Edit Online](#)

OEMs and mobile operators can each preload a set of custom ringtone files on Windows mobile devices, and they can set a default ringtone.

Ringtones should be a maximum of 150 KB and have a length of 5 to 15 seconds. They must be in .wma format, with a compression of 128 kbit/s for stereo or 64 kbit/s for mono. Partner ringtones should play at an appropriate volume relative to other sounds and ringtones, and there should be minimal distortion from the device speaker, at full volume.

## **Limitations and restrictions:**

- The names of the ringtones must be localized for all display languages that ship on the device.
- Partners must not move, delete, or modify the ringtones provided by Microsoft.
- Partners can only change the default alert sound used for phone calls – all other ringtone and alert defaults must not be changed unless specified elsewhere in the documentation.
- Users can delete partner ringtones.

Partners must keep the following design considerations in mind when implementing additional ringtones:

**File size:** Ringtone recommended maximum 150 KB; Alarm 100 KB(others as small as possible)

**Format:** .wma

**Compression:** WMA (128 kbps/stereo; 64 kbps/mono)

**Sound length:** Ringtones 5-15 seconds; Alarm 5-15 seconds; Calendar 1-3 seconds; Alerts 0.5-1.5 seconds

**Appropriate volumes:** Sounds should be appropriately balanced with ringtones and system sounds that ship with the OS.

**Minimal distortion** from device speaker, at full volume.

**Constraints:** FirstVariationOnly

## **Instructions:**

### **To add additional ringtones**

1. Create a .dll that contains the ringtone display name. For more information on how to do this, see [Create a resource-only .dll for localized strings](#).
2. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="AdditionalRingtones"
    Description="Use to add ringtone sound files and set a new default ringtone."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Localization/MUI">
            <!-- Use to add your base MUI DLL file -->
            <Asset Name="BaseDll" Source="" />

            <!-- Use to specify the language MUI packages (*.dll.mui) for the languages you are supporting and have localized strings for -->
            <Asset Name="LanguageDll/${langid}" Source="" />
            <Asset Name="LanguageDll/${langid}" Source="" />
            <Asset Name="LanguageDll/${langid}" Source="" />
            <!-- Add as many as you need -->
        </Settings>

        <Settings Path="EventSounds">
            <!-- Use to add additional ringtones -->
            <Asset Name="Ringtones" DisplayName="@DisplayStrings.dll,-Offset" Source="" Type="" />
            <Asset Name="Ringtones" DisplayName="@DisplayStrings.dll,-Offset" Source="" Type="" />
        </Settings>

    </Static>

</ImageCustomizations>

```

3. Specify an `Owner`.
4. Add the resource-only .dll that contains the ringtone sounds' display names by setting the `BaseDll` asset to point to the location of your base MUI DLL file. For example: `C:\Path\DisplayStrings.dll`.
5. Add the language MUI packages (\*.dll.mui) for all the languages you are supporting and have localized strings for. To do this:

- Set the asset's `Name` to `LanguageDll/ ${langid}` where `$(langid)` corresponds to the language. For example: `LanguageDll/en-US`.
- Set the asset's `Source` to the location of the .dll.mui file for that language. For example: `C:\Path\en-us\DisplayStrings.dll.mui`.
- Repeat the previous steps for the other languages.

The following example shows the customization answer file entries for en-US, fr-CA, and es-MX languages:

```

<Asset Name="LanguageDll/en-US" Source="C:\Path\en-us\DisplayStrings.dll.mui" />
<Asset Name="LanguageDll/fr-CA" Source="C:\Path\fr-CA\DisplayStrings.dll.mui" />
<Asset Name="LanguageDll/es-MX" Source="C:\Path\es-MX\DisplayStrings.dll.mui" />

```

6. Add additional notification sounds by adding a `Ringtones` asset. To do this:
  - Set the asset's `Name` to `Ringtones`.
  - Set the `DisplayName` to the name of the resource-only .dll file and specify the string offset. Replace `DisplayStrings.dll` with the name of your .dll file and replace `Offset` with the correct offset for the localized string. For example: `@DisplayStrings.dll,-104`.

- Set **Source** to the full path to the custom ringtone sound on your development machine. For example: *C:\Path\MellowRingtone.wma*.
- Optionally, set **Type** to either **OEM** or **MobileOperator** to distinguish the type of asset. If you do not set the type, this defaults to OEM.
- Repeat the previous steps for any additional ringtone sounds.

If you are setting the default alarm sound in addition to adding other alarm sound files, see the *To set a new default ringtone* section.

### To set a new default ringtone

1. Create a customization answer file using the contents shown in the following code sample or use the sample AdditionalRingtones.xml file.

```
<Settings Path="EventSounds">
    <!-- Use to set a new default ringtone -->
    <Setting Name="DefaultRingtone" Value="" />
</Settings>
```

2. Set the **Value** to the desired default ringtone sound file name. For example: *MellowRingtone.wma*.

### Testing:

1. Flash the build containing this customization and multiple display languages to a mobile device.
2. Go to the **Ringtone** screen in **Settings**.
3. Verify all added custom ringtones are in the **Ringtone** drop-down list.
4. If a new default ringtone is set, verify the **Ringtone** drop-down list defaults to the specified ringtone.
5. Verify the custom ringtone names are correct for all display languages on the device.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Call drop and call waiting sounds

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can customize the call drop and call waiting sounds.

On devices that exhibit a loud snap or tone during a phone call, OEMs can customize these sounds to add 50 ms of silence and improve the user experience. When used for this purpose, OEMs must:

- Add a sound file that contains 50 ms of silence. The sound file must be in a .wma format.
- Set the registry settings to the name of the sound file. This removes the loud snap or tone heard during some phone calls.

**Constraints:** FirstVariationOnly

**Instructions:**

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>

<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="CallDropCallWaitingSounds"
    Description="Use to customize the call drop and call waiting sounds."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="EventSounds">
            <!-- Use to add the OEM sound file. Change the Source path to point to the location and file name
            of the .wma sound file. -->
            <Asset Name="NotificationSounds" Source="" />
            <Asset Name="NotificationSounds" Source="" />

            <!-- Use to set the default call drop sound. Set the value to the file name of the sound file. -->
            <Setting Name="DefaultCallDropSound" Value="" />

            <!-- Use to set the default call waiting sound. Set the value to the file name of the sound file.
            -->
            <Setting Name="DefaultCallWaitingSound" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Add additional notification sounds by adding a `NotificationSounds` asset. To do this:

- Set the asset's `Name` to `NotificationSounds`.
- Set `Source` to the full path to the call drop sound file on your development machine. For example:  
`C:\Path\CallDropSound.wma`.
- Repeat the previous steps for any additional ringtone sound, such as if you are using a different sound file for the call waiting sound.

4. Set the value of `DefaultCallDropSound` to the desired default ringtone sound file name. For example:  
*CallDropSound.wma*.
5. Set the value of `DefaultCallWaitingSound` to the desired default ringtone sound file name. For example:  
*CallWaitingSound.wma*.

**Testing:**

1. Flash the build that contains this customization to a device.
2. Receive or make a phone call.
3. Verify that a loud snap or tone is no longer heard.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Camera shutter sound

10/2/2018 • 2 minutes to read • [Edit Online](#)

The camera shutter sound that occurs when the user takes a picture or starts filming a video can be turned off by removing the **Camera shutter** option from the **Sounds** settings screen.

This customization affects all camera apps on the mobile device. If camera sounds are not enforced, camera sounds will respond to the ringer and notification volume and sounds will not play through the combo device.

## Limitations and restrictions:

- OEMs can remove this user setting only for markets in which the camera shutter sound is a legally required component due to privacy laws.

**Constraints:** None

## Instructions:

- Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="CameraShutterSound"
    Description="Use to turn off the camera shutter sound by removing the Camera
shutter toggle
from the Settings CPL."
    Owner=""
    OwnerType="OEM">

<Static>

    <Settings Path="Camera">
        <Setting Name="ShutterSoundUI" Value="0" />
    </Settings>

</Static>

</ImageCustomizations>
```

- Specify an `Owner`.

- Set the `ShutterSoundUI``Value` to one of the following:

VALUE	DESCRIPTION
0 or 'Hide'	Hides the <b>Camera shutter</b> option from the <b>Sounds</b> settings screen.  When set to this value, the enforced shutter sound is on. This value enforces playback of camera sounds and sound will play through the combo device when the wired headset is plugged in.

VALUE	DESCRIPTION
1 or 'Show'	<p>Shows the <b>Camera shutter</b> option from the <b>Sounds</b> settings screen. This is the default.</p> <p>When set to this value, the enforced shutter sound is off.</p>

#### Testing:

1. Flash an image containing this customization to a phone.
2. Go to the **Sounds** settings screen.
3. Scroll down and verify that the **Camera shutter** setting is no longer visible.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Ringtone store application

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partner apps can be used to sell ringtones to users. The app owner must provide the service for the ringtone catalog and to manage downloads. Users are shown an option to **Get more** ringtones in the ringtone picker, from which they can automatically launch the ringtone store application.

**Constraints:** FirstVariationOnly

## Instructions:

1. Create an application that supports ringtones. For more information on how to do this, see *How to use the save ringtone task for Windows Phone* in the Windows Phone SDK Documentation.
2. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="RingtoneStoreApp"
    Description="Use to enable users to automatically launch the ringtone store
application that was created by
    the partner to sell ringtones to users."
    Owner=""
    OwnerType="OEM">

    <Static>

        <!-- Preload the ringtone store app. Specify the source, license, and ProvXML files. -->
        <Applications>
            <Application Source=""
                License=""
                ProvXML="" />
        </Applications>

        <Settings Path="EventSounds">
            <Setting Name="GetMoreRingtonesLink" Value="app://" />
        </Settings>

    </Static>

</ImageCustomizations>
```

3. Specify an `Owner`.
4. Preload your ringtone store app. To do this:
  - a. Set `Source` to the location and file name of your .xap or .appx. For example, `C:\Path\ContosoRingtoneStoreApp.xap`.
  - b. Set `License` to the location and name of the app license file. For example, `C:\Path\ContosoRingtoneStoreAppLicense.xml`.
  - c. Set `ProvXML` to the location and name of the provXML file. For example, `C:\Path\mpap_oemapp_01.provxml`.
5. Set the value of `GetMoreRingtonesLink` your application ID preceded by the `app://` prefix. For example, if your app ID is `{5B04B775-356B-4AA0-AAF8-6491FFEA5605}`, you must set the value to

`app://5B04B775-356B-4AA0-AAF8-6491FFEA5605`. You may also set it to  
`app://5B04B775-356B-4AA0-AAF8-6491FFEA5605/_default`.

## Testing:

1. Flash the build containing this customization to a phone.
2. Go to the **Sounds** screen in **Settings**.
3. Select the **Ringtone** picker.
4. In the ringtone picker screen, scroll to the bottom and verify that the **Get more** link is visible.
5. Tap the link and confirm that it launches your ringtone store application.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Customizations for SMS and MMS

10/2/2018 • 6 minutes to read • [Edit Online](#)

Contains settings that you can configure for SMS and MMS.

## In this section

TOPIC	DESCRIPTION
<a href="#">Add encoding extension tables for SMS</a>	Partners can extend the set of supported SMS encodings.
<a href="#">Automatic send retry and resize settings for MMS messages</a>	For MMS messages that have photo attachments and that fail to send, partners can choose to automatically resize the photo and attempt to resend the message.
<a href="#">Automatically retry downloading MMS messages</a>	Partners can configure the messaging app to automatically retry downloading an MMS message if the initial download attempt fails.
<a href="#">Content location in the multimedia message service center (MMSC)</a>	For networks that require it, partners can specify the default GET path within the MMSC to use when the GET URL is missing from the WAP push MMS notification.
<a href="#">Delay for resend attempts</a>	If an SMS message fails to send correctly, partners can specify the number of additional attempts and the minimum delay between them in seconds.
<a href="#">Disable editing of the SMS center number</a>	To meet market or mobile operator requirements, OEMs can configure a setting to prevent users from editing the <b>SMS center number</b> in the messaging settings CPL.
<a href="#">Disable the EMS long messages feature</a>	Partners can disable the enhanced messaging service (EMS), which concatenates text messages so that the user can enter more than 160 characters in a single message.
<a href="#">Expiration time for SMS messages</a>	Partners can set the expiration time before the device deletes the received parts of a long SMS message.
<a href="#">Extract phone numbers in strings</a>	Partners can extend the entity extraction feature that recognizes phone numbers in text. This customization allows strings of numbers that are concatenated to a string to be recognized.
<a href="#">Full error messages for SMS and MMS</a>	Partners can choose to display additional content in the conversation view when an SMS or MMS message fails to send.
<a href="#">IMS retry</a>	For networks that support it, when an outgoing SMS message fails to send due to a transient error, partners can specify the threshold for the number of attempts to resend the SMS over IMS before switching over to 3GPP or 3GPP2.

Topic	Description
IMSI authentication	For networks that require it, MMS messages can include the IMSI in the GET and POST header that the message center uses to authenticate the mobile subscriber.
International assisted dialing for SMS	If partners have turned off <a href="#">Assistance for dialing international phone numbers</a> , partners may still override the MCC and MNC used for plus code replacement when sending SMS.
Maximum length for SMS messages	Partners can specify a maximum length for SMS messages.
Maximum number of attachments for MMS messages	Partners can specify the maximum number of attachments for MMS messages, from 1 to 20.
Maximum number of recipients for SMS and MMS	Partners can set the maximum number of recipients to which a single SMS or MMS message can be sent.
MMS APN settings	Partners can choose to display an <b>Add MMS APN</b> or <b>Edit MMS APN</b> button that enables the user to configure the APN used for MMS.
MMS automatic download	Partners can choose to display an <b>Automatically download MMS</b> toggle to allows users to turn off automatic downloads of MMS messages. If the toggle is displayed, partners can also change the default value to stop automatic MMS downloads.
MMS data options	Partners can configure the MMS data options to show the <b>Allow MMS if cellular data is off</b> toggle in the <b>Messaging</b> settings screen, allow MMS messaging even if data is turned off, and allow MMS messaging even if data is turned off and the user is roaming.
MMS for group messages	For the setting that determines if group messages sent to multiple people must be sent as MMS, partners can customize the setting by hiding or showing the <b>Group Text</b> toggle in the <b>Messaging</b> settings screen, changing the default value, and configuring the option to alert the user of possible additional charges for sending a group text as MMS.
MMS receipt acknowledgement	Partners can specify whether the device automatically sends a receipt acknowledgment for MMS messages when messages arrive, and allow users to control the receipt acknowledgments by using the <b>Send MMS acknowledgement</b> toggle in the <b>Messaging</b> setting screen.
Permanent SMS message failures	Partners can mark SMS message failures as permanent failures so that the user will not be given the option to attempt to resend the SMS.
Ports that accept cellular broadcast messages	Partners can specify one or more ports from which the device will accept cellular broadcast messages.
Proxy authorization for MMS	Partners can specify the use of NAI information as a dedicated header for MMS authentication for mobile networks that require this functionality. The string value must be the MMS header used for authentication.

Topic	Description
Select multiple recipients for SMS and MMS messages	Partners can show the <b>select all contacts/unselect all</b> menu option to allow users to easily select multiple recipients for an SMS or MMS message.
Send SMS messages to SMTP addresses	Partners can configure SMS messages to be sent to email addresses as well as phone numbers.
Server for MMS acknowledgement messages	By default, the MMS transport sends an acknowledgement to the provisioned MMS application server (MMSC). However, on some networks, the correct server to use is sent as a URL in the MMS message. In that case, a registry key must be set, or else the acknowledgement will not be received and the server will continue to send duplicate messages.
SMS delivery confirmation	Partners can specify whether users receive notification that SMS messages could not be delivered, and determine whether users can control these notifications by using the <b>SMS delivery confirmation</b> toggle in the <b>Messaging</b> settings screen.
SMS encoding	Partners can change the default GSM 7-bit code page decoding and encoding, and can also extend the set of supported SMS encodings by setting an <b>always on</b> GSM 7-bit shift table, adding encoders, and adding decoders.
SMS intercept deny list	OEMs can specify one or more filters in order to intercept incoming SMS messages intended for mobile operator partner applications that are not installed on the device.
SMS intercept ports	OEMs can configure ports on which a Wireless Application Protocol (WAP)-formatted message can be intercepted by the mobile operator app.
Support HTTP cache-control no-transform for MMS	For networks that require it, OEMs can add support for the HTTP header Cache-Control No-Transform directive for MMS messages.
Supported protocols for service indication messages	Partners can add additional supported protocols for service indication messages.
Switch from SMS to MMS for long messages	For networks that do support MMS and do not support segmentation of SMS messages, partners can specify an automatic switch from SMS to MMS for long messages.
Truncated content handling for WAP push notification	For networks that require non-standard handling of single-segment incoming MMS WAP Push notifications, partners can specify that MMS messages may have some of their content truncated and that they may require special handling to reconstruct truncated field values.
Use insert-address-token or local raw address	To meet certain mobile operator requirements, OEMs can customize the OS image to use either the insert-address-token or the local raw address for the <b>From</b> field in MMS messages.

Topic	Description
<a href="#">Use UTF-8 for MMS messages with unspecified character encoding</a>	Some incoming MMS messages may not specify a character encoding. To properly decode MMS messages that do not specify a character encoding, OEMs can set UTF-8 to decode the message.
<a href="#">User agent profile for MMS messages</a>	Partners can specify a user agent profile to use on the device for MMS messages.
<a href="#">User agent string for MMS messages</a>	Partners can replace the entire user agent string for MMS.
<a href="#">User alert for service indication messages</a>	Partners can hide the user prompts for signal-medium messages.
<a href="#">Video attachments in MMS</a>	Partners can specify the transcoding to use for video files sent as attachments in MMS messages.
<a href="#">Voicemail SMS intercept</a>	Partners can define a filter that intercepts an incoming SMS message and triggers visual voicemail synchronization. The filtered message does not appear in the user's conversation list.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Add encoding extension tables for SMS

10/2/2018 • 4 minutes to read • [Edit Online](#)

Partners can extend the set of supported SMS encodings.

This is done by extending the following areas:

- Set an “always-on” GSM 7-bit shift table
- Add encoders
- Add decoders

To add additional National Language Shift Tables to the encoding, OEMs must replace Microsoft’s SMS provider by building your own code page DLL and setting it as the default. This section contains steps that OEMs can use to build an SMS provider with custom extension tables. The OEM is responsible for testing this code and their additions to it.

**Constraints:** None

This customization supports: **per-device** value

## Instructions:

The following steps describe how to configure and build the custom encodings:

1. Implement your code page DLL and make sure that it exports the [NlsDllCodePageTranslation](#) function.

### NOTE

Ignore the note about not using the function found on the MSDN web site. This note does not apply for SMS encoding.

When implementing your code page DLL, the DLL must have at least the following:

- A .def file that declares the name of the DLL—for example, `MyCodePageDLL`—and exports the **NlsDllCodePageTranslation** function:

```
LIBRARY      MyCodePageDLL

EXPORTS
    NlsDllCodePageTranslation
```

- A .c file that defines the DLL entry point function—for example, `DllMain`—as well as the **NlsDllCodePageTranslation** function. This is the function that the APIs will call in case a particular code page value is associated with your code page DLL.

```

// NlsDllCodePageTranslation
//
// This routine is the main exported procedure for the functionality in
// this DLL. All calls to this DLL must go through this function.
//
DWORD NlsDllCodePageTranslation(
    __in DWORD CodePage,
    __in DWORD dwFlags,
    __in_ecount(cchMultiByte) LPSTR lpMultiByteStr,
    __in int cchMultiByte,
    __out_ecount(cchWideChar) LPWSTR lpWideCharStr,
    __in int cchWideChar,
    __inout LPCPINFO lpCPIInfo)

```

- Inside the **NlsDllCodePageTranslation** function, you must:
  - Make sure that the *CodePage* value is one that you expect to handle.
  - If it is, proceed.
  - If it is not, call `SetLastError(ERROR_INVALID_PARAMETER)` and `return 0` to exit. Returning 0 from **NlsDllCodePageTranslation** indicates to the caller that an error occurred.
- Switch on the `dwFlags` to handle the cases for these values (defined in winnlsp.h):
  - `NLS_CP_CPIINFO`
  - `NLS_CP_CPIINFOEX`
  - `NLS_CP_MBTOWC`
  - `NLS_CP_WCTOMB`

Here's an example. Be sure to complete the items marked "OEM-TODO".

```

switch (dwFlags & 0xF0000000) // only look at the highest nibble
{
    case (NLS_CP_CPIINFO):
    {
        if (lpCPIInfo == NULL)
        {
            SetLastError(ERROR_INVALID_PARAMETER);
            return (0);
        }
        memset(lpCPIInfo, 0, sizeof(CPIINFO));

        // OEM-TODO: fill other parts of the CPIINFO structure as needed,
        // with one requirement for our test code:
        lpCPIInfo->DefaultChar[0] = 0x20;
        return (TRUE);
    }
    break;

    case (NLS_CP_CPIINFOEX): // this is actually optional
    {
        if (lpCPIInfo == NULL)
        {
            SetLastError(ERROR_INVALID_PARAMETER);
            return (0);
        }
        memset(lpCPIInfo, 0, sizeof(CPIINFOEX));

        // OEM-TODO: fill other parts of the CPIINFO structure as needed,
        // with one requirement for our test code:
        lpCPIInfo->DefaultChar[0] = 0x20;
    }
}

```

```

        return (TRUE);
    }
    break;

    case ( NLS_CP_MBTOWC ) :
    {
        // ensure unsupported flag combinations are not passed in
        if (dwFlags & ~(NLS_CP_MBTOWC | MB_ERR_INVALID_CHARS))
        {
            // other flags not allowed
            SetLastError(ERROR_INVALID_FLAGS);
            return (0);
        }

        // if caller expects us to figure out the input string length, do it now
        if (cchMultiByte == -1)
        {
            // see if the string is too long
            if (FAILED(StringCchLengthA(lpMultiByteStr, STRSAFE_MAX_CCH, (size_t *)(&cchMultiByte))))
            {
                SetLastError(ERROR_INVALID_PARAMETER);
                return (0);
            }
            // add one for the NULL terminator
            cchMultiByte += 1;
        }

        // OEM-TODO: convert lpMultiByteStr to lpWideCharStr according to own
        // conversion table, taking into account:
        // * dwFlags & MB_ERR_INVALID_CHARS (preserve MB_ERR_INVALID_CHARS flag
        //     in case the caller wants to error out on invalid input characters)
        // * cchWideChar - if cchWideChar == 0 or lpWideCharStr == NULL, that
        //     means the caller just wants to know how big lpWideCharStr should be

        return (the number of characters in converted lpWideCharStr + 1 for the NULL
terminator);
    }
    break;

    case ( NLS_CP_WCTOMB ) :
    {
        // ensure unsupported flag combinations are not passed in
        if (dwFlags & ~(NLS_CP_WCTOMB | WC_ERR_INVALID_CHARS))
        {
            // other flags not allowed
            SetLastError(ERROR_INVALID_FLAGS);
            return (0);
        }

        // if caller expects us to figure out the input string length, do it now
        if (cchWideChar == -1)
        {
            // see if the string is too long
            if (FAILED(StringCchLengthW(lpWideCharStr, STRSAFE_MAX_CCH, (size_t *)(&cchWideChar))))
            {
                SetLastError(ERROR_INVALID_PARAMETER);
                return (0);
            }
            // add one for the NULL terminator
            cchWideChar += 1;
        }

        // OEM-TODO: convert lpWideCharStr to lpMultiByteStr according to own
        // conversion table, taking into account:
        // * dwFlags & MB_ERR_INVALID_CHARS (preserve MB_ERR_INVALID_CHARS flag
        //     in case the caller wants to error out on invalid input characters)
        // * cchMultiByte - if cchMultiByte == 0 or lpMultiByteStr == NULL, that
        //     means the caller just wants to know how big lpMultiByteStr should be
    }
}

```

```

        return (the number of characters in converted lpMultiByteStr + 1 for the NULL
terminator);
    }
    break;
}
//
// if we haven't returned out of this function yet, the caller passed in an invalid flag
//
SetLastError(ERROR_INVALID_FLAGS);
return (0);
}
// end of NlsDllCodePageTranslation function

```

2. Pick a code page ID in the range 55050–55099.

Here are the code page identifiers for the new supported SMS encodings in Windows 10 Mobile as well as the reserved ranges to be used for future SMS encodings:

NLS CODE PAGE ID	DESCRIPTION
55000	GSM 7-bit
55001	GSM with Single Shift for Spanish
55002	GSM with Single Shift for Portuguese
55003	GSM with Single Shift for Turkish
55004	SMS Greek Reduction
55005–55049	Reserved
<b>55050–55099</b>	Available for OEM-supplied SMS encodings

3. Add your code page DLL to the device. To do this, set the **CodepageDLL** asset source to the location and file name of your code page DLL. For example:

```
<Asset Name="CodePageDLL" Source="C:\OEMCodePages\CodePage55050.dll" />
```

4. Register your DLL with NLS by setting the corresponding **EncodingCodePages/CodepageID550XX** setting.

For example, if the DLL is **Codepage55050.dll**, set **EncodingCodePages/CodepageID55050** as shown in the following example:

```
<Setting Name="EncodingCodePages/CodepageID55050" Value="CodePage55050.dll" />
```

5. Add the registered codepages to the custom OEM package as described in [SMS encoding](#).

Alternatively, you can write a settings app that dynamically sets the registry values depending on the user-selected encoding scheme. For example, if the OEM app sets the encoding to 55050, set the **Encodings/GSM7BitEncodingPage** setting and reboot the device.

```
<Setting Name="Encodings/GSM7BitEncodingPage" Value="55050" />
```

**Testing steps:**

Work with your mobile operator to test this customization on their network.

# Automatic send retry and resize settings for MMS messages

10/2/2018 • 3 minutes to read • [Edit Online](#)

For MMS messages that have photo attachments and that fail to send, partners can choose to automatically resize the photo and attempt to resend the message.

When this feature is enabled, partners must specify a size greater than or equal to 10 KB to use when resizing the photo.

Partners can also specify the number of times that the device can retry sending the failed MMS message and photo before the user receives a notification that the photo could not be sent.

The resize and retry settings can be used independently:

- If only the resize setting is set, the MMS will resize the photo and retry to send the message only once.
- If both settings are set, the MMS with a photo will be resized and the message will be resent up to three times.
- If only the retry setting is set, the MMS with the photo will not be resized and the message will be resent up to three times.

## **Limitations and restrictions:**

- This resize feature only applies to photos. Videos, audio files, and other file types will not be resized.
- The maximum number of automatic resend attempts is set to 3.

## **Constraints:** None

This customization supports: **per-SIM** value

## **Instructions:**

1. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="AutoResizeforMMS"
    Description="For MMS messages with photo attachments that fail to send, use to
    resize the photo and attempt to resend the MMS."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Messaging/PerSimSettings/$(__ICCID)">

            <!-- Specify the maximum size to use to resize the photo in KB. Minimum is 0xA (10 KB). -->
            <Setting Name="RetrySize" Value="" />

            <!-- Specify the number of times the MMS transport will attempt to resend the MMS, max is 0x3. -->
            <Setting Name="MaxRetryCount" Value="" />

        </Settings>
    </Variant>
</ImageCustomizations>

```

2. Specify an `Owner`.
3. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
4. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
5. Specify the `RetrySize``Value` to set a maximum size, in KB, to use when resizing the photo.

The minimum message retry size is 0xA (10 KB). If this number is less than 0xA, the value will be ignored and the device will not attempt to resize and resend large photos.

6. Specify the `MaxRetryCount``Value` to specify the number of times the MMS transport will attempt resending the MMS message. This value has a maximum limit of 0x3.

Keep the following in mind when setting the value for `RetrySize` and `MaxRetryCount`:

- If `MaxRetryCount` is not set and `RetrySize` is set, the MMS transport will retry sending the MMS message

once using the specified `RetrySize`. This behavior is similar to the default behavior in Windows 10 Mobile.

- If `MaxRetryCount` is set and `RetrySize` is not set, the MMS transport will not resize the MMS message and the message will be resent up to three times.

#### Testing steps:

1. Flash the build containing this customization to a device.
2. Go to the **messaging** application and attempt to attach a file that is larger than the limit that you set.
3. Send the photo. You may notice a slight delay. When the message arrives, the photo's size should be equal to the limit you specified for `RetrySize`.
4. If the message fails to send the first time, verify that the number of attempts to resend the message is equal to the limit you set for `MaxRetryCount`.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Automatically retry downloading MMS messages

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can configure the messaging app to automatically retry downloading an MMS message if the initial download attempt fails.

When this customization is enabled, the download is retried 3 times at 20-, 40-, and 60-second intervals. The following example shows how the retry intervals work using a random download time:

TIME	ACTIVITY
00:00:00	Initial download starts
00:00:11	Initial download fails due to a transient error. First wait starts and is scheduled for 20 seconds.
00:00:31	First wait times out, first retry download starts.
00:00:49	First retry download fails due to a transient error. Second wait starts and is scheduled for 40 seconds.
00:01:29	Second wait times out, second retry download starts.
00:01:34	Second retry download fails due to a transient error. Third wait starts and is scheduled for 60 seconds.
00:02:34	Third wait times out, third retry download starts.

If the MMS download fails after the third retry attempt, the message persists in the appropriate thread with a link that the user can tap to retry the download manually. If the user's manual download attempt fails, the automatic retries are triggered again.

## Constraints:

None  
This customization supports: **per-SIM** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="AutoRetryDownloadForMMS"
    Description="Use to configure the messaging app to automatically retry downloading
    an MMS message if the initial download attempt fails."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Define the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Messaging/PerSimSettings/$(__ICCID)">
            <!-- Set to 1 to enable or 0 to disable -->
            <Setting Name="AutoRetryDownload" Value="" />
        </Settings>
    </Variant>
    </ImageCustomizations>

```

2. Specify an **Owner**.
3. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
4. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
5. Set **AutoRetryDownload** to one of the following values:

VALUE	DESCRIPTION
1 or 0x1	Enable automatically retry downloading MMS messages.
0 or 0x0 Or if the <b>AutoRetryDownload</b> setting is missing	Disables automatically retry downloading MMS messages.

#### Testing steps:

1. Flash the build containing this customization to a device with a UICC or CDMA connection.
2. Successfully testing this customization requires the MMS message to fail, so work with mobile operator partner to test this customization on their network.

Be sure to test the scenario where the automatic retry attempt fails for a third time to verify the appearance of the manual download link.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Content location in the multimedia message service center (MMSC)

10/2/2018 • 2 minutes to read • [Edit Online](#)

For networks that require it, partners can specify the default GET path within the MMSC to use when the GET URL is missing from the WAP push MMS notification.

**Constraints:** None

This customization supports: **per-SIM** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="ContentInMMSC"
    Description="Use to specify the default GET path within the MMSC to use when the
    GET URL is missing."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Messaging/PerSimSettings/${__ICCID}">
            <Setting Name="DefaultContentLocationUrl" Value="" />
        </Settings>

    </Variant>
</ImageCustomizations>
```

2. Specify an **Owner**.

3. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and

SPN.

4. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
5. Set `DefaultContentLocationUrl``Value` to specify the default GET path within the MMSC.

**Testing steps:**

Work with your mobile operator to test this customization on their network.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Delay for resend attempts

10/2/2018 • 2 minutes to read • [Edit Online](#)

If an SMS message fails to send correctly, partners can specify the number of additional attempts and the minimum delay between them in seconds.

These settings are managed by the modem. For more information, contact the SoC vendor.

# Disable editing of the SMS center number

10/2/2018 • 2 minutes to read • [Edit Online](#)

To meet market or mobile operator requirements, OEMs can configure a setting to prevent users from editing the **SMS center number** in the messaging settings CPL.

By default, the setting does not exist and users can edit the **SMS center number**. A warning statement related to changing the SMS center number is also displayed below the SMS center number.

**Constraints:** None

This customization supports: **per-SIM** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>

<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="SmscPanelDisabled"
    Description="Use to prevent users from editing the 'SMS center number' in the
messaging settings CPL."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Messaging/PerSimSettings/$(__ICCID)">
            <Setting Name="SmscPanelDisabled" Value="" />
        </Settings>
    </Variant>
</ImageCustomizations>
```

2. Specify an  **Owner**.

3. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
4. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
5. Set `SmscPanelDisabled` to one of the following values:

VALUE	DESCRIPTION
1 or 'True'	Disables editing of the <b>SMS center number</b> and hides the warning statement.
0 or 'False'	Enables users to edit the <b>SMS center number</b> . This is the default behavior.

#### Testing steps:

1. Flash the build that contains this customization to a device.
2. Go to the **Messaging** settings screen.
3. Verify that the correct settings option is enabled depending on the default value that you set.
  - If `SmscPanelDisabled` is set to 1 or 'True', verify that the **SMS center number** cannot be edited.
  - If `SmscPanelDisabled` is set to 0 or 'False', verify that the **SMS center number** can be edited and the warning text below this option is visible.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Disable the EMS long messages feature

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can disable the enhanced messaging service (EMS), which concatenates text messages so that the user can enter more than 160 characters in a single message. If EMS is disabled, the user can still enter more than 160 characters. However, the send button is disabled and the UI alerts the user that the text message is too long instead of showing the character count of the text entry.

**Constraints:** None

This customization supports: **per-IMSI** value, **per-device** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="DisableEMSLongMessages"
    Description="Use to disable the enhanced messaging service (EMS) long messages
feature on Windows Phones. If EMS is disabled,
    users can still enter more than 160 characters, but the send button is
disabled and the user sees an alert
    that the message is too long."
    Owner=""
    OwnerType="OEM">

<!-- Use for the per-IMSI case

&lt;!-- Define the Targets --&gt;
&lt;Targets&gt;
    &lt;Target Id=""&gt;
        &lt;TargetState&gt;
            &lt;Condition Name="" Value="" /&gt;
            &lt;Condition Name="" Value="" /&gt;
        &lt;/TargetState&gt;
    &lt;/Target&gt;
&lt;/Targets&gt;

&lt;Static&gt;
    &lt;Settings Path="Multivariant"&gt;
        &lt;Setting Name="Enable" Value="1" /&gt;
    &lt;/Settings&gt;
    &lt;Settings Path="AutoDataConfig"&gt;
        &lt;Setting Name="Enable" Value="0" /&gt;
    &lt;/Settings&gt;
&lt;/Static&gt;

&lt;!-- Specify the Variant --&gt;
&lt;Variant Name=""&gt;
    &lt;TargetRefs&gt;
        &lt;TargetRef Id="" /&gt;
    &lt;/TargetRefs&gt;

    &lt;Settings Path="CellCore/PerIMSI/$(__IMSI)/SMS"&gt;
        &lt;!-- Set the value to 1 to limit the size of the message to one page and disable EMS. --&gt;
        &lt;Setting Name="SmsPageLimit" Value="1" /&gt;
    &lt;/Settings&gt;
&lt;/Variant&gt;

--&gt;

<!-- Use for the per-device case

&lt;Static&gt;
    &lt;Settings Path="CellCore/PerDevice/SMS"&gt;
        &lt;!-- Set the value to 1 to limit the size of the message to one page and disable EMS. --&gt;
        &lt;Setting Name="SmsPageLimit" Value="1" /&gt;
    &lt;/Settings&gt;
&lt;/Static&gt;

--&gt;

&lt;/ImageCustomizations&gt;
</pre>

```

2. Specify an `Owner`.
3. Determine if you need to use the **per-IMSI** or **per-device** setting.

For the **per-IMSI** case:

- a. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
  - b. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
4. Set `SmsPageLimit` to 1 to limit the size of the message to one page and disable EMS.

#### Testing:

1. Flash the build containing this customization to a device with a cellular connection.
2. Open the messaging application and attempt to send a message with a length that exceeds 160 characters.
3. Verify that the send button is disabled and that an alert that the text message is too long is showing next to the character count of the text entry.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Expiration time for SMS messages

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can set the expiration time before the device deletes the received parts of a long SMS message.

For example, if the device is waiting for a three-part SMS message and the first part has been received, the first part will be deleted when the time expires and the other part of the message has not arrived. If the second part of the message arrives before the time expires, the first and second parts of the message will be deleted if the last part does not arrive after the time expires. The expiration time is reset whenever the next part of the long message is received.

**Constraints:** None

This customization supports: **per-IMSI** value, **per-device** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample or use the sample SMSExpirationTime.xml file.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="SMSEExpirationTime"
    Description="Use to set the expiration time before the device deletes the received
parts of a long SMS message."
    Owner=""
    OwnerType="OEM">

<!-- Use for the per-IMSI case

    &lt!-- Define the Targets --&gt;
    &lt;Targets&gt;
        &lt;Target Id=""&gt;
            &lt;TargetState&gt;
                &lt;Condition Name="" Value="" /&gt;
                &lt;Condition Name="" Value="" /&gt;
            &lt;/TargetState&gt;
        &lt;/Target&gt;
    &lt;/Targets&gt;

    &lt;Static&gt;
        &lt;Settings Path="Multivariant"&gt;
            &lt;Setting Name="Enable" Value="1" /&gt;
        &lt;/Settings&gt;
        &lt;Settings Path="AutoDataConfig"&gt;
            &lt;Setting Name="Enable" Value="0" /&gt;
        &lt;/Settings&gt;
    &lt;/Static&gt;

    &lt!-- Specify the Variant --&gt;
    &lt;Variant Name=""&gt;
        &lt;TargetRefs&gt;
            &lt;TargetRef Id="" /&gt;
        &lt;/TargetRefs&gt;

        &lt;Settings Path="CellCore/PerIMSI/$(__IMSI)/SMS"&gt;
            &lt;Setting Name="MessageExpirySeconds" Value="" /&gt;
            &lt;!-- Default is 0x15180 which is 1 day or 86400 seconds. --&gt;
        &lt;/Settings&gt;

    &lt;/Variant&gt;
    --&gt;

    &lt!-- Use for the per-device case

    &lt;Static&gt;
        &lt;Settings Path="CellCore/PerDevice/SMS"&gt;
            &lt;Setting Name="MessageExpirySeconds" Value="" /&gt;
            &lt;!-- Default is 0x15180 which is 1 day or 86400 seconds. --&gt;
        &lt;/Settings&gt;
    &lt;/Static&gt;

    --&gt;
&lt;/ImageCustomizations&gt;</pre>

```

2. Specify an **Owner**.

3. Determine if you need to use the **per-IMSI** or **per-device** setting.

For the **per-IMSI** case:

- Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
- Define settings for a **Variant**, which are applied if the associated target's conditions are met.

4. Set `MessageExpirySeconds` to the number seconds that the device should wait before deleting the received parts of a long SMS messages. This value should be in hexadecimal and must be prefixed with 0x.

The default value is 0x15180, which is equivalent to 1 day or 86,400 seconds.

#### Testing:

1. Flash the build containing this customization to a device with a UICC.
2. From another device, send a long SMS message to the device containing the customization.
3. Verify that received parts of the long message are deleted within the expiration time that you have set if the next part is not received within that same amount of time.

#### Note

Work with your mobile operator to fully test this customization.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Extract phone numbers in strings

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can extend the entity extraction feature that recognizes phone numbers in text. This customization allows strings of numbers that are concatenated to a string to be recognized.

Windows supports entity extraction in the Messaging app (which is also shared with the Email and Calendar apps). This feature can detect a sequence of numbers, which can be a phone number, in a received SMS or MMS message and enable the user to call the phone number by making it a target that the user can tap and easily save, or call without retyping the number.

This customization extends entity extraction by enabling OEMs to specify whether a sequence of numbers that is concatenated to a string (in a left-to-right device language including English, French, Italian, Simplified Chinese, Traditional Chinese, and other left-to-right languages) should be detected as a phone number. The minimum amount of numbers that the OS considers as a phone number is five (5).

When this feature is enabled, only the numeric sequence is underlined and shown as a tappable string; the non-numeric character is not underlined. When this feature is enabled on devices that have dual SIMs, it is applied to both SIM slots.

**Constraints:** FirstVariationOnly

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="ExtractPhoneNumbersInStrings"
    Description="Use to tag any 5-or-more digit number as a phone number that users can
    tap even when
                                there is no space between the string and the number."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Messaging/GlobalSettings">
            <Setting Name="ExtractPhoneNumbersInStrings" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an  .

3. Set  `ExtractPhoneNumbersInStrings` to one of the following values:

VALUE	DESCRIPTION
-------	-------------

VALUE	DESCRIPTION
0 or 'False'	<p>Specifies that the OS should not detect a sequence of five or more digits concatenated to a string as a tappable phone number.</p> <p>For example, when the value is set to 0 or 'False', a sequence of digits such as 55512 is detected as a phone number, but P55512 or ABC55512EFG are not.</p>
1 or 'True'	<p>Specifies that the OS should detect a sequence of five or more digits concatenated to a string as a tappable phone number.</p> <p>For example, when the value is set to 1 or 'True', a sequence of digits such as 55512 is detected as a phone number while the sequence '55512' in the strings P55512 or ABC55512EFG are also detected as a phone number.</p>

### Testing steps:

Work with your mobile operator partner to fully test this customization on their network.

1. Flash the build containing this customization to a device with at least one active SIM or UICC.
2. Make sure that the device language is set to a language that has left-to-right characters, such as English, French, Italian, Simplified Chinese, and Traditional Chinese.
3. Open the Messaging app.
4. Receive several SMS or MMS messages that contain a sequence of digits that are:
  - a. Less than 5 digits
  - b. More than 5 digits
  - c. Less than 5 digits and have the digits concatenated to one or more non-numerical characters, such as P1234 or PhoneNumber1234.
  - d. More than 5 digits and with the digits concatenated to one or more non-numerical characters, such as P55512 or PhoneNumber55512.
5. For each case specified in the previous step, verify whether the sequence of numbers is detected as a phone number by the OS based on the value you specified for the `ExtractPhoneNumbersInStrings` setting.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Full error messages for SMS and MMS

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can choose to display additional content in the conversation view when an SMS or MMS message fails to send. This content includes a specific error code in decimal format that the user can report to technical support. Common errors also include a friendly string to help the user self-diagnose and fix the problem.

**Constraints:** None

This customization supports: **per-SIM** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="SMSErrorMessage"
    Description="Use to display additional content in the conversation view when an SMS
or MMS message fails to send."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Messaging/PerSimSettings/$(__ICCID)">
            <Setting Name="ErrorCodeEnabled" Value="" />
        </Settings>
    </Variant>
</ImageCustomizations>
```

2. Specify an **Owner**.
3. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.

4. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
5. The `LimitRecipients` setting limits the maximum number of recipients that a user can send messages to, and this value is in the range  $0 < \text{LimitRecipients} \leq 500$  (decimal). When setting the value for `LimitRecipients`, you can use either decimal or the equivalent hexadecimal value (with a `0x` prefix).

Set `ErrorCodeEnabled` to one of the following values:

VALUE	DESCRIPTION
1 or 'True'	Displays the error messages with an explanation of the problem and the decimal-format error codes.
0 or 'False'	Does not display the full error message.

#### Testing steps:

1. Flash the build containing this customization to a device.
2. Ensure that the device is able to send SMS or MMS messages.
3. Open the messaging application to send SMS or MMS messages.
4. Work with your mobile operator to create error scenarios for messaging.
5. The error message displayed on the messaging screen should start with an explanation in words of the problem, and end in either "MMS error: #####" or "SMS error: #####".

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# IMS retry

10/2/2018 • 2 minutes to read • [Edit Online](#)

For networks that support it, when an outgoing SMS message fails to send due to a transient error, partners can specify the threshold for the number of attempts to resend the SMS over IMS before switching over to 3GPP or 3GPP2. Partners can also specify to retry sending the SMS message once over 3GPP or 3GPP2 without a user prompt if the original message was sent over IMS and failed.

**Constraints:** None

This customization supports: **per-IMSI** value, **per-device** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="IMSRetry"
    Description="Use to specify settings related to SMS messages sent over IMS."
    Owner=""
    OwnerType="OEM">

    <!-- Use for the per-IMSI case

        <!-- Define the Targets -->
        <Targets>
            <Target Id="">
                <TargetState>
                    <Condition Name="" Value="" />
                    <Condition Name="" Value="" />
                </TargetState>
            </Target>
        </Targets>

        <Static>
            <Settings Path="Multivariant">
                <Setting Name="Enable" Value="1" />
            </Settings>
            <Settings Path="AutoDataConfig">
                <Setting Name="Enable" Value="0" />
            </Settings>
        </Static>

        <!-- Specify the Variant -->
        <Variant Name="">
            <TargetRefs>
                <TargetRef Id="" />
            </TargetRefs>

            <Settings Path="CellCore/PerIMSI/$(__IMSI)/SMS">
                <!-- Use to specify the threshold for the number of attempts to resend the SMS over IMS -->
                <Setting Name="3GPP/IMS/AttemptThresholdForIMS" Value="" />

                <!-- Use to retry sending the SMS message once without a user prompt -->
                <Setting Name="3GPP/IMS/RetryEnabled" Value="" />
            </Settings>
        </Variant>
    <!-->

    <!-- Use for the per-device case

        <Static>
            <Settings Path="CellCore/PerDevice/SMS">
                <!-- Use to specify the threshold for the number of attempts to resend the SMS over IMS -->
                <Setting Name="3GPP/IMS/AttemptThresholdForIMS" Value="" />

                <!-- Use to retry sending the SMS message once without a user prompt -->
                <Setting Name="3GPP/IMS/RetryEnabled" Value="" />
            </Settings>
        </Static>
    <!-->

</ImageCustomizations>
```

2. Specify an `Owner`.
3. Determine if you need to use the **per-IMSI** or **per-device** setting.

For the **per IMSI** case:

- a. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
  - b. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
4. Set the `3GPP/IMS/AttemptThresholdForIMS``Value` to specify the threshold for the number of attempts to resend the SMS over IMS. If you need to enable SMS over IMS, then you must set the value to 1 or higher integer.
- For example, if you set this value to 2, this will result in two total attempts. If the value is set to N, it will result in N number of attempts. This value will vary for each mobile operator so work with your mobile operator partner to obtain the correct or required value. If the threshold is exceeded, the SMS will no longer be sent over IMS, but will be sent using the available 3GPP or 3GPP2 channel.
5. Set the `3GPP/IMS/RetryEnabled``Value` to retry sending the SMS message once without a user prompt if the original message was sent over IMS and failed with RIL\_E\_IMSTEMPFAILURE. The client is not notified of the initial failure. The retry attempt will not be over IMS.

The default value is 1, which means this behavior is enabled by default. Set to 0 to disable this behavior.

#### Testing:

1. Flash the build containing this customization to a device with a UICC or CDMA connection.
2. Successfully testing this customization requires the message to fail so please work with your mobile operator to test this customization on their network.

During testing, resending the failed SMS message requires the UI. Verification of the tests need to be done using the logs at the network and modem levels.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# IMSI authentication

10/2/2018 • 2 minutes to read • [Edit Online](#)

For networks that require it, MMS messages can include the IMSI in the GET and POST header that the message center uses to authenticate the mobile subscriber.

**Constraints:** None

This customization supports: **per-SIM** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="IMSIAuthentication"
    Description="Use to include the IMSI in the GET and POST header used by the message
    center to authenticate the mobile subscriber."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Messaging/PerSimSettings/$(__ICCID)">
            <!-- The string value must be set to the IMSI provided by the UICC -->
            <Setting Name="ImsiAuthenticationToken" Value="" />
        </Settings>

    </Variant>
    </ImageCustomizations>
```

2. Specify an **Owner**.
3. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.

4. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
5. Set the `ImsiAuthenticationToken``Value` to the token used as the header for authentication. The string value should match the IMSI provided by the UICC.

**Testing steps:**

Work with your mobile operator to test this customization on their network.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# International assisted dialing for SMS

10/2/2018 • 2 minutes to read • [Edit Online](#)

If partners have turned off [Assistance for dialing international phone numbers](#), partners may still override the MCC and MNC used for plus code replacement when sending SMS. These values are used to generate the correct IDD. This change applies regardless of roaming status. By setting `AssistedDialingMcc` and `AssistedDialingMnc`, international assisted dialing will be enabled for SMS if the user setting for international assisted dialing is enabled.

For devices that support IMS over SMS, partners can override support for the assisted dialing plus (+) code for SMS by setting `AssistedDialingPlusCodeSupportOverride`. If enabled, the OS will not convert the plus (+) code to the proper assisted number when the user turns on the dialing assist option.

**Constraints:** None

This customization supports: **per-SIM** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="SMSAssistedDialing"
    Description="Use to override the MCC and MNC used for sending SMS messages."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Messaging/PerSimSettings/$(__ICCID)">
            <Setting Name="AssistedDialingMcc" Value="" />
            <Setting Name="AssistedDialingMnc" Value="" />

            <!-- For an IMS over SMS supported device, partners can override the assisted dialing plus code
            support for SMS.
            <Setting Name="AssistedDialingPlusCodeSupportOverride" Value="" />
            -->
        </Settings>
    </Variant>
</ImageCustomizations>

```

2. Specify an `Owner`.
3. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
4. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
5. Set the values for the following settings to enable international assistance dialing for SMS:

SETTING	DESCRIPTION
<code>AssistedDialingMcc</code> : REG_SZ	The Mobile Country Code (MCC) to use for sending SMS.
<code>AssistedDialingMnc</code> : REG_SZ	The Mobile Network Code (MNC) to use for sending SMS.

6. For a device that supports IMS over SMS, you can override the support for the assisted dialing plus (+) code for SMS. To do this, set `AssistedDialingPlusCodeSupportOverride` to one of the following values:

VALUE	DESCRIPTION
0 or 'False'	Don't override the assisted dialing plus code support. This is the default OS value.
1 or 'True'	Override the assisted dialing plus code support. The OS will not convert the plus (+) code to the proper assisted number when the user turns on the dialing assist option.

**Testing steps:**

Work with your mobile operator to test this customization on their network.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Maximum length for SMS messages

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can specify a maximum length for SMS messages. This requires setting both the maximum number of SMS fragments per SMS message, from 1 to 255, and the maximum size in bytes of each SMS fragment, from 16 to 140 bytes.

**Constraints:** None

This customization supports: **per-IMSI** value, **per-device** value

**Instructions:**

1. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="MaxLengthSMS"
    Description="Use to configure the maximum length for SMS messages."
    Owner=""
    OwnerType="OEM">

    <!-- Use for the per-IMSI case

        <!-- Define the Targets -->
        <Targets>
            <Target Id="">
                <TargetState>
                    <Condition Name="" Value="" />
                    <Condition Name="" Value="" />
                </TargetState>
            </Target>
        </Targets>

        <Static>
            <Settings Path="Multivariant">
                <Setting Name="Enable" Value="1" />
            </Settings>
            <Settings Path="AutoDataConfig">
                <Setting Name="Enable" Value="0" />
            </Settings>
        </Static>

        <!-- Specify the Variant -->
        <Variant Name="">
            <TargetRefs>
                <TargetRef Id="" />
            </TargetRefs>
            <Settings Path="CellCore/PerIMSI/$(__IMSI)/SMS">
                <Setting Name="SmsFragmentLimit" Value="" />
                <Setting Name="SmsPageLimit" Value="" />
            </Settings>
        </Variant>
    <!-->

    <!-- Use for the per-device case

        <Static>
            <Settings Path="CellCore/PerDevice/SMS">
                <Setting Name="SmsFragmentLimit" Value="" />
                <Setting Name="SmsPageLimit" Value="" />
            </Settings>
        </Static>
    <!-->

</ImageCustomizations>
```

2. Specify an **Owner**.

3. Determine if you need to use the **per-IMSI** or **per-device** setting.

For the **per-IMSI** case:

- Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
- Define settings for a **Variant**, which are applied if the associated target's conditions are met.

4. Use **SmsFragmentLimit** to set the maximum number of bytes in the user data body of an SMS message. You

must set the value between 16 (**0x10**) and 140 (**0x8C**).

5. Use `SmsPageLimit` to set the maximum number of segments in a concatenated SMS message. You must set the value to 255 (**0xFF**) or smaller.

#### Testing:

1. Flash the build containing this customization to a device that contains a UICC or a configured CDMA connection.
2. Open the messaging application and attempt to send a message with a length that exceeds the combination of `SmsFragmentLimit`  $\times$  `SmsPageLimit`.

You should receive an error message indicating that the message was too long.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Maximum number of attachments for MMS messages

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can specify the maximum number of attachments for MMS messages, from 1 to 20. The default is 5.

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="MaxNumberAttachmentsForMMS"
    Description="Use to specify the maximum number of attachments for MMS messages
    (from 1 to 20)."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Messaging/PerSimSettings/$(__ICCID)">
            <Setting Name="MMSLimitAttachments" Value="" />
        </Settings>
    </Variant>
    </ImageCustomizations>
```

2. Specify an `Owner`.
3. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
4. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
5. Set `MMSLimitAttachments` to a value from 1 to 20 to specify the default number of attachments for MMS messages. The default set by the OS is 5.

## Testing:

1. Flash the build containing this customization to a device.
2. Go to the messaging application and attempt to attach multiple items up to and greater than the limit.
3. The **attach** button should become disabled after the maximum number of attachments has been reached.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Maximum number of recipients for SMS and MMS

10/2/2018 • 3 minutes to read • [Edit Online](#)

Partners can set the maximum number of recipients to which a single SMS or MMS message can be sent.

The maximum number of recipients that a user can send an SMS or MMS message to is limited to 500. This limit exists because the OS also supports the [Select multiple recipients for SMS and MMS messages](#) feature and having the number of recipients for SMS or MMS messages limited to a large, but finite, number ensures that there is no system performance degradation or negative impact to the user experience. The maximum number of recipients that a user can send messages to is in the range 0 <= 500 (decimal).

**Constraints:** None

This customization supports: **per-SIM** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

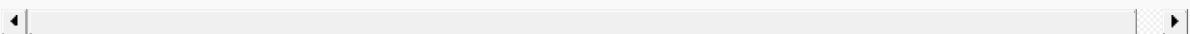
```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="LimitMessagingMaxRecipients"
    Description="Use to set the maximum number of recipients for SMS or MMS messages."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Messaging/PerSimSettings/$(__ICCID)">
            <Setting Name="LimitRecipients" Value="" />
        </Settings>
    </Variant>
</ImageCustomizations>
```



2. Specify an **Owner**.
3. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
4. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
5. The **LimitRecipients** setting limits the maximum number of recipients that a user can send messages to, and this value is in the range  $0 < \text{LimitRecipients} \leq 500$  (decimal). When setting the value for **LimitRecipients**, you can use either decimal or the equivalent hexadecimal value (with a 0x prefix).

Set the value for **LimitRecipients** according to the following rules:

- If **LimitRecipients** is not set, the maximum number of recipients defaults to 500.
- If **LimitRecipients** is set to a value greater than 500, the maximum number of recipients is set to 500.
- If **LimitRecipients** is set to 1, the maximum number of recipients is set to 1 and the multi-select button in the single select screen is disabled
- If **LimitRecipients** is set to a value between 2 and 500, the maximum number of recipients is equal to the number that was set

## Instructions:

### Testing:

1. Flash the build containing this customization to a device with a UICC or network connection.
2. Make sure your device contains more contacts than the number you set for **LimitRecipients**.
3. Open the messaging application, create a new message, and tap + to add recipients.
4. If [Select multiple recipients for SMS and MMS messages](#) is not enabled, the behavior for **LimitRecipients** is the same as in Windows Phone 8.
5. If [Select multiple recipients for SMS and MMS messages](#) is enabled, tap the multi-select menu option, tap ..., choose **select all contacts** and verify that you can see the following message:

#### Too many contacts

**You can select up to X contacts. If you select all, you will have Y contacts.**

X is the decimal equivalent of the value that you set for **LimitRecipients**. Y is the total number of contacts you have selected.

6. If [Select multiple recipients for SMS and MMS messages](#) is enabled, tap the multi-select menu option, individually tap the names of your contacts and select one more than the number you set for **LimitRecipients**. Verify that you can see the following message:

#### Too many contacts

**You can select up to X contacts, and you already have that many.**

X is the decimal equivalent of the value that you set for **LimitRecipients**.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# MMS APN settings

10/2/2018 • 3 minutes to read • [Edit Online](#)

Partners can choose to display an **add mms apn** or **edit mms apn** button that enables the user to configure the APN used for MMS. APN values entered by the user are not verified and may not work. The user-entered APN is always used if available; it is not overridden by ADC or over the air updates.

## Limitations and restrictions:

- If partners have set a list of permitted push proxy gateways, any user-entered APN that does not match with a PPG value in the list will fail to work.

## Constraints:

None

This customization supports: **per-IMSI** value, **per-device** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="CellularSettings"
    Description="Use to display an 'add mms apn' or 'edit mms apn' button to enable
    users to configure the APN used for MMS."
    Owner=""
    OwnerType="OEM">

    <!-- Use for the per-IMSI case

        <!-- Define the Targets -->
        <Targets>
            <Target Id="">
                <TargetState>
                    <Condition Name="" Value="" />
                    <Condition Name="" Value="" />
                </TargetState>
            </Target>
        </Targets>

        <Static>
            <Settings Path="Multivariant">
                <Setting Name="Enable" Value="1" />
            </Settings>
            <Settings Path="AutoDataConfig">
                <Setting Name="Enable" Value="0" />
            </Settings>
        </Static>

        <!-- Specify the Variant -->
        <Variant Name="">
            <TargetRefs>
                <TargetRef Id="" />
            </TargetRefs>

            <Settings Path="CellCore/PerIMSI/$(__IMSI)/CellUX">
                <!-- Hides or shows the 'add mms apn' button in the SIM settings page.
                    Set to 0 or 'No' (to show) or set to 1 or 'Yes' (to hide). -->
                <Setting Name="HideMMSAPN" Value="" />

                <!-- Hides or shows the 'IP type' setting in the MMS APN settings screen.
                    Set to 0 or 'No' (to show) or set to 1 or 'Yes' (to hide). -->
            </Settings>
        </Variant>
    </Static>
</ImageCustomizations>
```

```

        Set to 0 or 'No' (to show) or set to 1 or 'Yes' (to hide). -->
<Setting Name="HideMMSAPNIPType" Value="" />

        <!-- Changes the default IP type. Set to 0 or 'IPv4' (for IPv4), 1 or 'IPv6' (for IPv6), or 2 or
        'IPv4v6' (for IPv4v6). -->
        <Setting Name="MMSAPNIPTypeIfHidden" Value="" />
    </Settings>
</Variant>

-->

<!-- Use for the per-device case

<Static>
    <Settings Path="CellCore/PerDevice/CellUX">
        <!-- Hides or shows the 'add mms apn' button in the SIM settings page.
        Set to 0 or 'No' (to show) or set to 1 or 'Yes' (to hide). -->
        <Setting Name="HideMMSAPN" Value="" />

        <!-- Hides or shows the 'IP type' setting in the MMS APN settings screen.
        Set to 0 or 'No' (to show) or set to 1 or 'Yes' (to hide). -->
        <Setting Name="HideMMSAPNIPType" Value="" />

        <!-- Changes the default IP type. Set to 0 or 'IPv4' (for IPv4), 1 or 'IPv6' (for IPv6), or 2 or
        'IPv4v6' (for IPv4v6). -->
        <Setting Name="MMSAPNIPTypeIfHidden" Value="" />
    </Settings>
</Static>

-->

</ImageCustomizations>

```

2. Specify an **Owner**.

3. Determine if you need to use the **per-IMSI** or **per-device** setting.

For the **per-IMSI** case:

- a. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
- b. Define settings for a **Variant**, which are applied if the associated target's conditions are met.

4. To hide or show the **add mms apn** button in the SIM settings screen: Set the value for **HideMMSAPN** to one of the following:

VALUE	DESCRIPTION
0 or 'No'	Shows the <b>add mms apn</b> button in the <b>SIM</b> settings screen.
1 or 'Yes'	Hides the <b>add mms apn</b> button in the <b>SIM</b> settings screen.

5. To hide or show the **IP type** setting in the MMS APN settings screen: Set the value for **HideMMSAPNIPType** to one of the following:

VALUE	DESCRIPTION
0 or 'No'	Shows the <b>IP type</b> setting in the MMS APN settings screen.
1 or 'Yes'	Hides the <b>IP type</b> drop-down in the MMS APN settings screen.

6. To change the default IP type shown in the **IP type** settings drop-down: Set the value for `MMSAPNIPTypeIfHidden` to one of the following:

VALUE	DESCRIPTION
0 or 'IPV4'	Sets the default IP type to IPv4.
1 or 'IPV6'	Sets the default IP type to IPv6.
1 or 'IPV4V6'	Sets the default IP type to IPv4v6.

#### Testing:

1. Flash the build containing this customization to a device.
2. If the **add mms apn** button is configured to be hidden:
  - Go to the **Messaging** settings screen and verify that the **add mms apn** button is no longer visible.
  - Go to the **Cellular & SIM** settings screen and verify that the **add mms apn** button is no longer visible.
3. If the **add mms apn** button is not configured to be hidden, tap the button and verify that the **IP type** setting either shows the correct default value or is hidden.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# MMS automatic download

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can choose to display an **Automatically download MMS** toggle to allows users to turn off automatic downloads of MMS messages. If the toggle is displayed, partners can also change the default value to stop automatic MMS downloads.

**Constraints:** None

This customization supports: **per-SIM** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="MMSAutomaticDownload"
    Description="Use to display an 'Automatically download MMS' toggle to allow users
    to turn off auto downloads of MMS messages."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Messaging/PerSimSettings/$(__ICCID)/AutomaticallyDownload">
            <Setting Name="ShowAutomaticallyDownloadMMSToggle" Value="" />
            <Setting Name="AutomaticallyDownload" Value="" />
        </Settings>
    </Variant>

```

2. Specify an **Owner**.
3. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.

- Define settings for a **Variant**, which are applied if the associated target's conditions are met.
- To hide or show the **Automatically download MMS** toggle in the messaging settings screen, set the value of `ShowAutomaticallyDownloadMMSToggle` to one of the following:

VALUE	DESCRIPTION
0 or 'False'	Hides the toggle.
1 or 'True'	Shows the toggle.

- To set the default value of the **Automatically download MMS** toggle, set the value of `AutomaticallyDownload` to one of the following:

VALUE	DESCRIPTION
0 or 'False'	Sets the default value to off.
1 or 'True'	Sets the default value to on. This is the OS default value.

#### Testing:

- Flash the build containing this customization to a device.
- Go to the **Messaging** settings screen.
- Verify that the **Automatically download MMS** toggle is visible and has the expected default value.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# MMS data options

10/2/2018 • 2 minutes to read • [Edit Online](#)

For some phones, the cellular data plan includes the data used to send MMS. If cellular data is turned off, so is the ability to send MMS messages. However, other data plans bill MMS data separately. In this case, it's necessary to have a setting that allows for MMS messages to be sent even if the data toggle is off.

Partners can configure the MMS data options to:

- Show the **Allow MMS if cellular data is off** toggle in the **Messaging** settings screen.
- Allow MMS messaging even if data is turned off.
- Allow MMS messaging even if data is turned off and the user is roaming.

OEMs can configure these settings on both single SIM and C+G dual SIM phones.

**Constraints:** None

This customization supports: **per-SIM** value

**Instructions:**

1. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="AllowMMSIfDataIsOff"
    Description="Use to configure MMS settings if data is turned off."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Messaging/PerSimSettings/$(__ICCID)/AllowMmsIfDataIsOff">
            <Setting Name="AllowMmsIfDataIsOffSupported" Value="" />
            <Setting Name="AllowMmsIfDataIsOff" Value="" />
            <Setting Name="AllowMmsIfDataIsOffWhileRoaming" Value="" />
        </Settings>
    </Variant>
</ImageCustomizations>

```

2. Specify an `Owner`.
3. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
4. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
5. To hide or show the **Allow MMS if cellular data is off** toggle in the **Messaging** settings screen, set the value of `AllowMmsIfDataIsOffSupported` to one of the following:

VALUE	DESCRIPTION
0 or 'False'	Hides the <b>Allow MMS if cellular data is off</b> toggle.
1 or 'True'	Shows the <b>Allow MMS if cellular data is off</b> toggle. This is the default OS value.

6. Before you set the value for `AllowMmsIfDataIsOff`, note that if you do not set `ExemptFromDisablePolicy` to 1 (0 by default), then you must:

- Hide the **Allow MMS if cellular data is off** toggle by setting `AllowMmsIfDataIsOffSupported` to 0 (1 by default).
- Set `AllowMmsIfDataIsOff` itself to 1 (0 by default).

For more information about `ExemptFromDisablePolicy`, see [CM\\_CellularEntries CSP](#).

To allow or disallow MMS messaging even if data is turned off, set the value of `AllowMmsIfDataIsOff` to one of the following:

VALUE	DESCRIPTION
0 or 'False'	MMS is on when data is on, and off when data is off. MMS will also be off when roaming if the user has set the phone to not use data while roaming.  This is the default OS value.
1 or 'True'	MMS is on even when the data toggle is off, but not when roaming if data is off.

7. To allow MMS if data is turned off while the user is roaming, set value of `AllowMmsIfDataIsOffWhileRoaming` to one of the following:

VALUE	DESCRIPTION
0 or 'False'	MMS messaging is off when roaming even if the user has set to allow MMS messaging if data is turned off.
1 or 'True'	MMS messaging is on when roaming even while data is off.  Shows the user alert.

**\*\*Note\*\*** This setting is only visible if `AllowMmsIfDataIsOff` is set.

## Testing:

1. Flash the build containing this customization to a device.
2. Go to the **Messaging** settings screen.
3. Verify if the messaging options match the values you set for each setting.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# MMS delivery confirmation

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can specify whether users receive notification that MMS messages could not be delivered, and determine whether users can control this by using the **MMS delivery confirmation** toggle in the **Messaging** settings screen. By default, this user setting is visible but turned off.

**Constraints:** None

This customization supports: **per-SIM** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="MMSDeliveryConfirmation"
    Description="Use to specify whether users receive notification that MMS messages
could
not be delivered, and determine whether users can control this by
using the
'MMS delivery confirmation' toggle in the messaging settings screen."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Messaging/PerSimSettings/$(__ICCID)/RequestDeliveryReport">
            <Setting Name="RequestDeliveryReportIsSupported" Value="" />
            <Setting Name="RequestDeliveryReport" Value="" />
        </Settings>
    </Variant>
</ImageCustomizations>
```

2. Specify an `Owner`.
3. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
4. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
5. To hide or show the toggle for MMS delivery confirmation, set the value of `RequestDeliveryReportIsSupported` to one of the following:

VALUE	DESCRIPTION
0 or 'False'	Hides the toggle.
1 or 'True'	Shows the toggle.

6. To set the default value for the MMS delivery confirmation toggle, set the value of `RequestDeliveryReport` to one of the following:

VALUE	DESCRIPTION
0 or 'False'	Sets the default value to off.
1 or 'True'	Sets the default value to on.

### Testing:

1. Flash the build containing this customization to a device.
2. Go to the settings screen in the **Messaging** app.
3. Verify the **MMS delivery confirmation** toggle default value or check that it is no longer visible depending on the settings and values that you used.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# MMS for group messages

10/2/2018 • 2 minutes to read • [Edit Online](#)

For the setting that determines if group messages sent to multiple people must be sent as MMS, partners can customize the setting by hiding or showing the **Group text** toggle in the **Messaging** settings screen, changing the default value, and configuring the option to alert the user of possible additional charges for sending a group text as MMS.

**Constraints:** None

This customization supports: **per-SIM** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="MMSGroupText"
    Description="Use to determine if group messages sent to multiple people must be
sent as MMS.
Partners can show/hide the 'Group text' toggle, configure the default
value, and
alert users of possible additional charges for sending a group text."
Owner=""
OwnerType="OEM">

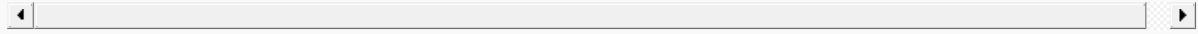
<!-- Define the Targets -->
<Targets>
    <Target Id="">
        <TargetState>
            <Condition Name="" Value="" />
            <Condition Name="" Value="" />
        </TargetState>
    </Target>
</Targets>

<Static>
    <Settings Path="Multivariant">
        <Setting Name="Enable" Value="1" />
    </Settings>
    <Settings Path="AutoDataConfig">
        <Setting Name="Enable" Value="0" />
    </Settings>
</Static>

<!-- Specify the Variant -->
<Variant Name="">
    <TargetRefs>
        <TargetRef Id="" />
    </TargetRefs>

    <Settings Path="Messaging/PerSimSettings/${__ICCID}/MMSGroupText">
        <Setting Name="ShowMMSGroupTextUI" Value="" />
        <Setting Name="MMSGroupText" Value="" />
        <Setting Name="ShowMmsGroupTextWarning" Value="" />
    </Settings>
</Variant>
</ImageCustomizations>

```



2. Specify an `Owner`.
3. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
4. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
5. To hide or show the toggle for **Group text** in the **Messaging** settings screen, set the value of `ShowMMSGroupTextUI` to one of the following:

VALUE	DESCRIPTION
0 or 'False'	Hides the toggle.
1 or 'True'	Shows the toggle.

6. To set the default value for the **Group text** toggle, set the value of `MMSGroupText` to one of the following:

VALUE	DESCRIPTION
0 or 'False'	Sets the default value to off.
1 or 'True'	Sets the default value to on.

7. To hide or show the warning that alerts users of possible additional charges before sending a group text as MMS, set value of `ShowMmsGroupTextWarning` to one of the following:

VALUE	DESCRIPTION
0 or 'False'	Hides the user alert.
1 or 'True'	Shows the user alert.

#### Testing:

1. Flash the build containing this customization to a device.
2. Go to the **Messaging** settings screen.
3. Verify that the **Group text** toggle is visible or not. Also verify that the toggle has the correct default value.
4. Write and then send a group text as MMS. Depending on the value you set for the user alert, verify if the user alert is shown or hidden.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# MMS receipt acknowledgement

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can specify whether the device automatically sends a receipt acknowledgment for MMS messages when messages arrive, and allow users to control the receipt acknowledgments by using the **Send MMS acknowledgement** toggle in the **Messaging** setting screen. By default, this user setting is visible and turned on.

**Constraints:** None

This customization supports: **per-SIM** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="MMSReceiptAcknowledgement"
    Description="Use to hide or show the 'Send MMS acknowledgement' toggle in Settings,
and configure the default
value for the toggle."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Messaging/PerSimSettings/$(__ICCID)/AllowSendingDeliveryReport">
            <Setting Name="AllowSendingDeliveryReportIsSupported" Value="" />
            <Setting Name="AllowSendingDeliveryReport" Value="" />
        </Settings>
    </Variant>
</ImageCustomizations>
```

2. Specify an **Owner**.
3. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and

SPN.

4. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
5. To hide or show the **Send MMS acknowledgement** toggle, set the value of `AllowSendingDeliveryReportIsSupported` to one of the following:

VALUE	DESCRIPTION
0 or 'False'	Hides the toggle.
1 or 'True'	Shows the toggle.

6. To set the default value for the **Send MMS acknowledgement** toggle, set the value of `AllowSendingDeliveryReport` to one of the following:

VALUE	DESCRIPTION
0 or 'False'	Sets the default value to off.
1 or 'True'	Sets the default value to on.

### Testing:

1. Flash the build containing this customization to a device.
2. Go to the **Messaging** settings screen.
3. Verify the **Send MMS acknowledgement** toggle default value or check that it is no longer visible depending on the registry key that you used.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Permanent SMS message failures

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can mark SMS message failures as permanent failures so that the user will not be given the option to attempt to resend the SMS.

**Constraints:** None

This customization supports: **per-IMSI** value, **per-device** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="PermanentSMSMessageFailures"
    Description="Use to mark SMS message failures as permanent failures so that users
    cannot attempt to resend the SMS."
    Owner=""
    OwnerType="OEM">

<!-- Use for the per-IMSI case

    &lt!-- Define the Targets --&gt;
    &lt;Targets&gt;
        &lt;Target Id=""&gt;
            &lt;TargetState&gt;
                &lt;Condition Name="" Value="" /&gt;
                &lt;Condition Name="" Value="" /&gt;
            &lt;/TargetState&gt;
        &lt;/Target&gt;
    &lt;/Targets&gt;

    &lt;Static&gt;
        &lt;Settings Path="Multivariant"&gt;
            &lt;Setting Name="Enable" Value="1" /&gt;
        &lt;/Settings&gt;
        &lt;Settings Path="AutoDataConfig"&gt;
            &lt;Setting Name="Enable" Value="0" /&gt;
        &lt;/Settings&gt;
    &lt;/Static&gt;

    &lt!-- Specify the Variant --&gt;
    &lt;Variant Name=""&gt;
        &lt;TargetRefs&gt;
            &lt;TargetRef Id="" /&gt;
        &lt;/TargetRefs&gt;

        &lt;Settings Path="CellCore/PerIMSI/$(__IMSI)/SMS"&gt;
            &lt;Setting Name="3GPP2/ErrorHandling/UseReservedAsPermanent" Value="" /&gt;
        &lt;/Settings&gt;
    &lt;/Variant&gt;
--&gt;

    &lt!-- Use for the per-device case

    &lt;Static&gt;
        &lt;Settings Path="CellCore/PerDevice/SMS"&gt;
            &lt;Setting Name="3GPP2/ErrorHandling/UseReservedAsPermanent" Value="" /&gt;
        &lt;/Settings&gt;
    &lt;/Static&gt;
--&gt;
&lt;/ImageCustomizations&gt;
</pre>

```

2. Specify an **Owner**.

3. Determine if you need to use the **per-IMSI** or **per-device** setting.

For the **per-IMSI** case:

- Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
- Define settings for a **Variant**, which are applied if the associated target's conditions are met.

4. Set the `3GPP2/ErrorHandling/UseReservedAsPermanent` to one of the following values:

VALUE	DESCRIPTION
1 or 'Yes'	Marks SMS failures as permanent. This disables the UI option that allows the user to attempt to resend the SMS message.
0 or 'No'	Does not mark SMS failures as permanent.

**Testing:**

1. Flash the build containing this customization to a device with a UICC.
2. Open the messaging application and attempt to send a message to a number that will result in an SMS failure.
3. Verify that the user option to resend the message does not show up.

**Note**

Work with your mobile operator to fully test this customization.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Ports that accept cellular broadcast messages

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can specify one or more ports from which the device will accept cellular broadcast messages.

**Constraints:** None

This customization supports: **per-SIM** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>

<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="SMSCellularBroadcastPorts"
    Description="Use to specify one or more ports from which the device will accept
cellular broadcast messages."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Messaging/PerSimSettings/${__ICCID}">
            <Setting Name="BroadcastChannels" Value="" />
        </Settings>
    </Variant>
</ImageCustomizations>
```

2. Specify an **Owner**.
3. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
4. Define settings for a **Variant**, which are applied if the associated target's conditions are met.

5. Set the `BroadcastChannels` value to the port number(s) that can accept cellular broadcast messages. For example, `1234;5678;9012` and so on.

If you specify the same port that Windows 10 Mobile already recognizes as an Emergency Alert port (a CMAS or ETWS port number) and a cell broadcast message is received on that port, the user will only receive the message once. The message that is received will be displayed as an Emergency Alert message.

**Testing:**

Work with your mobile operator to test this customization on their network.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Proxy authorization for MMS

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can specify the use of NAI information as a dedicated header for MMS authentication for mobile networks that require this functionality. The string value must be the MMS header used for authentication.

**Constraints:** None

This customization supports: **per-SIM** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="ProxyAuthorizationMMS"
    Description="Use to set the NAI information as a dedicated header for MMS
authentication."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Messaging/PerSimSettings/$(__ICCID)">
            <Setting Name="ProxyAuthorizationToken" Value="Proxy-Authorization:Basic" />
        </Settings>

    </Variant>
    </ImageCustomizations>
```

2. Specify an **Owner**.
3. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
4. Define settings for a **Variant**, which are applied if the associated target's conditions are met.

5. Do not change the `ProxyAuthorizationToken``Value`. `Proxy-Authorization` is the HTTP header and `Basic` denotes Basic64 encoding and not any other encoding.

**Testing steps:**

Work with your mobile operator to properly test this customization on their network.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Select multiple recipients for SMS and MMS messages

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can show the **select all contacts/unselect all** menu option to allow users to easily select multiple recipients for an SMS or MMS message. This menu option provides users with an easier way to add multiple recipients and may also meet a mandatory requirement for some mobile operator networks.

Windows 10 Mobile supports the following select multiple recipients feature:

- A multi-select chooser, which enables users to choose multiple contacts.
- A **select all contacts/unselect all** menu option, which enables users to select or unselect all their contacts. This option is not shown by default and must be enabled by the OEM.

## Note

Note that this feature will only be enabled if the `LimitRecipients` setting is set to more than 1 recipient. For more information, see [Maximum number of recipients for SMS and MMS](#).

## Constraints:

This customization supports: **per-SIM** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="SelectMultipleRecipients"
    Description="Use to enable users to easily select multiple recipients for SMS and
    MMS messages."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Messaging/PerSimSettings/${__ICCID}">
            <Setting Name="AllowSelectAllContacts" Value="" />
        </Settings>
    </Variant>
</ImageCustomizations>

```

2. Specify an `Owner`.
3. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
4. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
5. Set the value for `AllowSelectAllContacts` to one of the following:

VALUE	DESCRIPTION
0 or 0x0	The <b>select all contacts/unselect all</b> menu option is not shown.
1 or 0x1	The <b>select all contacts/unselect all</b> menu option appears in the app menu.

If `AllowSelectAllContacts` is not set or missing, the \*\*select all contacts\*\*/\*\*unselect all\*\* menu option is not shown.

### **Testing steps:**

1. Flash the build containing this customization to a device.
2. Open the **Messaging** application and create a new SMS or MMS message.
3. Select the + button to add message recipients.
4. Verify that the multi-select icon appears in the **Choose a contact** screen. If you have more than one contact, this icon will be active. Otherwise, it will appear grey or inactive.
5. Tap the multi-select icon then tap ... for more menu options.
6. Verify that you can see the **select all contacts** menu option.
7. Tap **select all contacts**.
8. Tap ... and verify that you can see the **unselect all** menu option.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Send SMS messages to SMTP addresses

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can configure SMS messages to be sent to email addresses as well as phone numbers.

**Constraints:** None

This customization supports: **per-SIM** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="SMSToSMTPShortCode"
    Description="Use to configure SMS messages to be sent to email addresses and phone
numbers."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Messaging/PerSimSettings/$(__ICCID)">
            <Setting Name="AllowSMSToSMTPAddress" Value="" />
            <Setting Name="SMSToSMTPShortCode" Value="" />
        </Settings>

    </Variant>
</ImageCustomizations>
```

2. Specify an **Owner**.
3. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
4. Define settings for a **Variant**, which are applied if the associated target's conditions are met.

5. Set the value for `AllowSMSToSMTPAddress` to one of the following values:

VALUE	DESCRIPTION
0 or 'False'	Disables sending SMS messages to SMTP addresses.
1 or 'True'	Enables sending SMS messages to SMTP addresses.

6. Set the `SMSToSMTPShortCode` value to the correct short code for your mobile operator. This value is the destination SMTP gateway phone number, and must be provided by the mobile operator.

#### Testing:

1. Flash the build containing this customization to a device that contains a SIM or network connection for your mobile operator.
2. Open the messaging app and attempt to send a message to a valid email address.
3. The message should send and arrive successfully.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Server for MMS acknowledgement messages

10/2/2018 • 2 minutes to read • [Edit Online](#)

By default, the MMS transport sends an acknowledgement to the provisioned MMS application server (MMSC). However, on some networks, the correct server to use is sent as a URL in the MMS message. In that case, a registry key must be set, or else the acknowledgement will not be received and the server will continue to send duplicate messages.

**Constraints:** None

This customization supports: **per-SIM** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>

<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="ServerMMSAcknowledgement"
    Description="Use to enable some networks to correctly acknowledge MMS messages."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Messaging/PerSimSettings/$(__ICCID)">
            <Setting Name="UseDefaultAddress" Value="" />
        </Settings>

    </Variant>
</ImageCustomizations>
```

2. Specify an **Owner**.

3. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and

SPN.

4. Define settings for a **Variant**, which are applied if the associated target's conditions are met.

5. Set `UseDefaultAddress` to one of the following values:

VALUE	DESCRIPTION
1 or 0x1	Enable some networks to correctly acknowledge MMS messages.
0 or 0x0	Disable the feature.

**Testing steps:**

Work with your mobile operator to test this customization on their network.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# SMS delivery confirmation

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can specify whether users receive notification that SMS messages could not be delivered, and determine whether users can control these notifications by using the **SMS delivery confirmation** toggle in the **Messaging** settings screen. By default, this user setting is not visible, and delivery confirmations are not turned on.

**Constraints:** None

This customization supports: **per-SIM** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>

<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="SMSDeliveryConfirmation"
    Description="Use to hide or show the 'SMS delivery confirmation' toggle in
    Settings, and configure the default
    value for the toggle."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Messaging/PerSimSettings/$(__ICCID)/SMSDeliveryNotify">
            <Setting Name="DeliveryNotifySupported" Value="" />
            <Setting Name="SMSDeliveryNotify" Value="" />
        </Settings>

    </Variant>
</ImageCustomizations>
```

2. Specify an **Owner**.

3. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
4. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
5. To hide or show the **SMS delivery confirmation** toggle, set the value of `DeliveryNotifySupported` to one of the following:

VALUE	DESCRIPTION
0 or 'False'	Hides the toggle.
1 or 'True'	Shows the toggle.

6. To set the default value for the **SMS delivery confirmation** toggle, set the value of `SMSDeliveryNotify` to one of the following:

VALUE	DESCRIPTION
0 or 'False'	Sets the default value to off.
1 or 'True'	Sets the default value to on.

### Testing:

1. Flash the build containing this customization to a device.
2. Go to the **Messaging** settings screen.
3. Verify the **SMS delivery confirmation** toggle default value or check that it is no longer visible depending on the registry key that you used.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# SMS encoding

10/2/2018 • 3 minutes to read • [Edit Online](#)

Partners can change the default GSM 7-bit code page decoding and encoding, and can also extend the set of supported SMS encodings by setting an "always-on" GSM 7-bit shift table, adding encoders, and adding decoders.

Using augmented NLS encodings, Windows 10 Mobile supports the following SMS encodings, as defined in the [3GPP TS 23.038](#) specification:

- 7-bit ASCII (used on 3GPP2 only)
- GSM 7-bit encoding
  - Default alphabet
  - GSM with Single Shift for Spanish
  - GSM with Single Shift for Portuguese
  - GSM with Single Shift for Turkish
- UCS2
- KSC 5601
- Shift-JIS
- SMS Greek Reduction
- Other binary data encoding usage

By default, Windows 10 Mobile supports the GSM 7-bit default alphabet table. Partners do not need to set a registry key to support this. However, partners can change the default GSM 7-bit code page to decode and encode all incoming and outgoing SMS messages by using `GSM7BitEncodingPage` and setting this to one of the allowed values. See the following instructions for more information on how to do this.

Partners can change the default GSM 7-bit encoding to one of the other supported SMS encodings by setting the appropriate registry setting. See the following instructions for more information on how to do this.

OEMs can also extend the set of supported SMS encodings by setting an "always-on" GSM 7-bit shift table, adding encoders, and adding decoders. For more information, see [Add encoding extension tables for SMS](#).

**Constraints:** None

This customization supports: **per-IMSI** value, **per-device** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="SMSEncoding"
    Description="Use to configure the SMS encoding settings."
    Owner=""
    OwnerType="OEM">

    <!-- Use for the per-IMSI case
        <!-- Define the Targets -->
        <Targets>
            <Target Id="">
                <TargetState>
                    <Condition Name="" Value="" />
                    <Condition Name="" Value="" />
                </TargetState>
            </Target>
        </Targets>

        <Static>
            <Settings Path="Multivariant">
                <Setting Name="Enable" Value="1" />
            </Settings>
            <Settings Path="AutoDataConfig">
                <Setting Name="Enable" Value="0" />
            </Settings>
        </Static>

        <!-- Specify the Variant -->
        <Variant Name="">
            <TargetRefs>
                <TargetRef Id="" />
            </TargetRefs>

            <Settings Path="CellCore/PerIMSI/$(__IMSI)/SMS">
                <Setting Name="Encodings/GSM7BitEncodingPage" Value="" />
                <Setting Name="Encodings/GSM8BitEncodingPage" Value="" />
                <Setting Name="Encodings/UseASCII" Value="" />
                <Setting Name="Encodings/UseKeyboardLanguage" Value="" />
                <Setting Name="Encodings/SendUDHNLSS" Value="" />
                <Setting Name="Encodings/OctetEncodingPage" Value="" />
            </Settings>
        </Variant>
    -->

    <!-- Use for the per-device case
        <Static>
            <Settings Path="CellCore/PerDevice/SMS">
                <Setting Name="Encodings/GSM7BitEncodingPage" Value="" />
                <Setting Name="Encodings/GSM8BitEncodingPage" Value="" />
                <Setting Name="Encodings/UseASCII" Value="" />
                <Setting Name="Encodings/UseKeyboardLanguage" Value="" />
                <Setting Name="Encodings/SendUDHNLSS" Value="" />
                <Setting Name="Encodings/OctetEncodingPage" Value="" />
            </Settings>
        </Static>
    -->
</ImageCustomizations>

```

2. Specify an **Owner**.

3. Determine if you need to use the **per-IMSI** or **per-device** setting.

For the **per-IMSI** case:

- Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.

- b. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
4. Configure the settings and values depending on the SMS encoding that you want to set.

SETTING NAME	DESCRIPTION																				
	CODE PAGE VALUE	SETTING VALUE	CODE PAGE																		
<b>Encodings/GSM7BitEncodingPage</b>	<p>Used to set the code page value for GSM 7-bit encoding.</p> <p>The possible values are as follows:</p> <table border="1"> <thead> <tr> <th>CODE PAGE VALUE</th> <th>SETTING VALUE</th> <th>CODE PAGE</th> </tr> </thead> <tbody> <tr> <td>55000</td> <td>0xD6D8</td> <td>Default alphabet</td> </tr> <tr> <td>55001</td> <td>0xD6D9</td> <td>GSM with Single Shift for Spanish</td> </tr> <tr> <td>55002</td> <td>0xD6DA</td> <td>GSM with Single Shift for Portuguese</td> </tr> <tr> <td>55003</td> <td>0xD6DB</td> <td>GSM with Single Shift for Turkish</td> </tr> <tr> <td>55004</td> <td>0xD6DC</td> <td>SMS Greek Reduction</td> </tr> </tbody> </table>			CODE PAGE VALUE	SETTING VALUE	CODE PAGE	55000	0xD6D8	Default alphabet	55001	0xD6D9	GSM with Single Shift for Spanish	55002	0xD6DA	GSM with Single Shift for Portuguese	55003	0xD6DB	GSM with Single Shift for Turkish	55004	0xD6DC	SMS Greek Reduction
CODE PAGE VALUE	SETTING VALUE	CODE PAGE																			
55000	0xD6D8	Default alphabet																			
55001	0xD6D9	GSM with Single Shift for Spanish																			
55002	0xD6DA	GSM with Single Shift for Portuguese																			
55003	0xD6DB	GSM with Single Shift for Turkish																			
55004	0xD6DC	SMS Greek Reduction																			
<b>Encodings/GSM8BitEncodingPage</b>	<p>Used to set the code page value for GSM 8-bit encoding.</p>																				
<b>Encodings/UseASCII</b>	<p>Used only for CDMA carriers that use 7-bit ASCII encoding instead of GSM 7-bit encoding.</p> <p>The possible values are as follows:</p> <table border="1"> <thead> <tr> <th>SETTING VALUE</th> <th>DESCRIPTION</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Turns on 7-bit ASCII</td> </tr> <tr> <td>0</td> <td>Turns off 7-bit ASCII</td> </tr> </tbody> </table>			SETTING VALUE	DESCRIPTION	1	Turns on 7-bit ASCII	0	Turns off 7-bit ASCII												
SETTING VALUE	DESCRIPTION																				
1	Turns on 7-bit ASCII																				
0	Turns off 7-bit ASCII																				

SETTING NAME	DESCRIPTION						
<b>Encodings/UseKeyboardLanguage</b>	<p>Used to change the SMS encoding based on the language of the device keyboard. This only works for these languages: Spanish, Portuguese, Turkish</p> <p>The possible values are as follows:</p> <table border="1"> <thead> <tr> <th>SETTING VALUE</th><th>DESCRIPTION</th></tr> </thead> <tbody> <tr> <td>1</td><td>Turns on SMS encoding based on keyboard language.</td></tr> <tr> <td>0</td><td>Turns off SMS encoding based on keyboard language.</td></tr> </tbody> </table>	SETTING VALUE	DESCRIPTION	1	Turns on SMS encoding based on keyboard language.	0	Turns off SMS encoding based on keyboard language.
SETTING VALUE	DESCRIPTION						
1	Turns on SMS encoding based on keyboard language.						
0	Turns off SMS encoding based on keyboard language.						
<b>Encodings/SendUDHNLSS</b>	<p>Used to specify whether outgoing SMS messages using GSM 7-bit encoding will contain header information that defines the shift table used.</p> <p>The possible values are as follows:</p> <table border="1"> <thead> <tr> <th>SETTING VALUE</th><th>DESCRIPTION</th></tr> </thead> <tbody> <tr> <td>1</td><td>Turns on the feature.</td></tr> <tr> <td>0</td><td>Turns off the feature.</td></tr> </tbody> </table>	SETTING VALUE	DESCRIPTION	1	Turns on the feature.	0	Turns off the feature.
SETTING VALUE	DESCRIPTION						
1	Turns on the feature.						
0	Turns off the feature.						
<b>Encodings/OctetEncodingPage</b>	Used to set the code page for octet (binary) encoding.						

### Testing:

1. Flash the build containing this customization to a device.
2. Go to the messaging application to write and send an SMS message.
3. Verify that the written SMS message used the correct encoding and, when possible, also verify that the sent SMS message was received with the correct encoding.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# SMS intercept deny list

10/2/2018 • 4 minutes to read • [Edit Online](#)

OEMs can specify one or more filters in order to intercept incoming SMS messages intended for mobile operator partner applications that are not installed on the device.

Applications from mobile operator partners can receive notifications from incoming SMS messages. Specifically, the Windows 10 Mobile SMS intercept feature takes string filters declared by the partner application in its manifest and matches these with the opening text in the SMS messages. If there is a string match, the message is delivered to the corresponding application instead of the Microsoft messaging application.

As part of the SMS intercept feature, OEMs can specify one or more filters in a *deny list* to be matched against incoming SMS messages intended for mobile operator partner applications that are not installed on the device. If an SMS message matches a filter on the deny list, the message is dropped and never delivered to the messaging application or shown to the user. If filters are not specified as part of the deny list and the mobile operator application is not installed on the device, the SMS message is delivered to the Microsoft messaging application and displayed to the user.

The SMS intercept deny list follows the following filter and string matching rules:

## Filter rules:

- Each filter can only be matched to one application ID. If multiple applications listed the same filter, the first application is used. This is not deterministic across boots.
- The longest filter strings are matched to ensure that exact matches are made before partial matches.
- A filter must be at least 3 characters long. If the filter is less than 3 characters long it will be ignored and omitted from the filter match list.
- A filter can be up to 74 characters long.
- A filter can only contain valid Unicode characters.
- Byte by byte comparison is used without consideration for culture.
- Comparisons are not case sensitive.

## String match rules:

- The string must start as the leading character of the message.
- The filter string is considered a match if it is an exact match or it is a partial match where the entire filter is contained in the first segment of the body of the message.

The SMS intercept deny list runs after the partner application string matching has been done.

The following examples show the results for the string match based on the preceding rules:

FILTER STRING	BODY OF THE SMS MESSAGE	STRING MATCH RESULT
//MOMessagingClient	//MOMessagingClient	Yes – exact match
MOMessagingClient	MOMessagingClient	Yes – exact match

FILTER STRING	BODY OF THE SMS MESSAGE	STRING MATCH RESULT
//MOMessagingClient	//MOMessagingClient12345 You can sign on!	Yes – partial match
//MOMessagingClient	You can sign on now! //MOMessagingClient	No – wrong location
//M*MessagingClient	//MOMessagingClient	No – no wildcards in the filter string
//	//MOMessagingClient	No – filter string must be at least 3 characters
//MOMessagingClient	//MOMessageClient You can sign on!	No – not an exact match
//MOMessagingClient	_//MOMessagingClient	No – not a leading string
MOMessagingClient	MOMessagingClient	No – leading whitespace or tab
//MOMessagingClient	//momessagingclient	Case insensitive
//MOMessagingClient //MO	//MOMessagingClient	Exact match (on //MOMessagingClient) over partial (//MO)

**Constraints:** None

This customization supports: **per-SIM** value

**Instructions:**

1. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="SMSInterceptDenyList"
    Description="Use to specify one or more filters in a deny list to be matched
against incoming SMS messages intended for
mobile operator partner applications that are not installed on the device."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Messaging/PerSimSettings/$(__ICCID)">
            <!-- Set the value for the ports where the device will accept cellular broadcast messages.
            The value must be in REG_MULTI_SZ format, such as "Prefix1;Prefix2;Prefix3" and so on -->
            <Setting Name="SmsInterceptPrefixes" Value="" />
        </Settings>

    </Variant>
</ImageCustomizations>

```

2. Specify an `Owner`.
3. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
4. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
5. Set the `SmsInterceptPrefixes` value to one or more filters in a deny list. For example, `Prefix1;Prefix2;Prefix3` and so on.

The following notes apply for this customization:

- SMS messages that start with the filters (designated by the placeholders `Prefix1,Prefix2`, and so on) are dropped and never delivered. Filters are provided by the mobile operators.
- The number of filters that OEMs can add in the deny list is not limited.
- The system does not back up the set of settings and values for this feature.
- The system only reads the deny list at boot time. If setting values are updated at a later time, the device must be restarted for the updated list to take effect.

**Testing steps:**

1. Flash the build containing this customization to a device.
2. Send several SMS messages to the device (some that match the filters and others that do not).
3. Based on the filters that you added, verify if the messages that match the filters were correctly dropped and verify that those that don't match any filters show up in the messaging app.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# SMS intercept ports

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can configure ports on which a Wireless Application Protocol (WAP)-formatted message can be intercepted by the mobile operator app.

Certain mobile operators require the ability to intercept SMS messages for processing by a mobile operator app rather than by the standard Microsoft messaging app. To meet these operator requirements, OEMs can specify string filters in a deny list to be matched against incoming SMS messages intended for operator partner apps that are not installed on the device. For more information on how to do this, see [SMS intercept deny list](#). In addition, operators can also require the ability to configure the port on which a Wireless Application Protocol (WAP)-formatted message can be intercepted by the mobile operator app. The incoming WAP message must have its destination port set to be one of the configured ports in order for the message to be accepted. To configure the correct port, OEMs can use the `SmsInterceptPorts` setting that's documented in this topic.

**Constraints:** None

This customization supports: **per-SIM** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="SMSInterceptPorts"
    Description="Use to specify one or more ports on which a WAP-formatted message can
be intercepted by a mobile operator app."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Messaging/PerSimSettings/$(__ICCID)">
            <Setting Name="SmsInterceptPorts" Value="" />
        </Settings>
    </Variant>
    </ImageCustomizations>

```

2. Specify an `Owner`.
3. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
4. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
5. Set the `SmsInterceptPorts` value to one or more ports on which a WAP-formatted message can be intercepted by the custom MO app. For example, `4100;04102;04456` and so on.

### **Caution**

Any port number can be configured except for 2948, which is the standard port of a WAP push.

### **Testing steps:**

Work with your mobile operator partner to correctly test this customization on their network.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Support HTTP cache-control no-transform for MMS

10/2/2018 • 2 minutes to read • [Edit Online](#)

For networks that require it, OEMs can add support for the HTTP header Cache-Control No-Transform directive for MMS messages. When set, proxies and transcoders are instructed not to change the HTTP header and the content should not be modified (`Cache-Control: no-transform\r\n`).

When this directive is not set or the registry setting is missing, the HTTP header is set to Cache-Control No-Cache (`Cache-Control: no-cache\r\n`).

**Constraints:** None

This customization supports: **per-SIM** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="SetCacheControlNoTransform"
    Description="Use to add support for the HTTP header Cache-Control No-Transform
directive for MMS messages."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Messaging/PerSimSettings/${__ICCID}">
            <Setting Name="SetCacheControlNoTransform" Value="1" />
        </Settings>

    </Variant>
</ImageCustomizations>
```

2. Specify an `Owner`.

3. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
4. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
5. Do not modify the `SetCacheControlNoTransform``Value`. A value of 1 or 0x1 adds support for the HTTP header Cache-Control No-Transform directive.

When the `SetCacheControlNoTransform``Value` is set to 0 or 0x0 or when the setting is not set, the default HTTP header Cache-Control No-Cache directive is used.

**Testing:**

Flash the build containing this customization to a device with a UICC.

**Warning**

To fully verify this customization, you will need to use a tool that captures HTTP traffic.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Supported protocols for service indication messages

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can add additional supported protocols for service indication messages. By default, service indication messages are valid only if their href begins with one of the four supported protocols: http, https, wsp, or wsps.

**Constraints:** None

This customization supports: **per-SIM** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="SIProtocols"
    Description="Use to add additional supported protocols for service indication
messages."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Messaging/PerSimSettings/$(__ICCID)">
            <Setting Name="SIProtocols" Value="" />
        </Settings>

    </Variant>
    </ImageCustomizations>
```

2. Specify an **Owner**.
3. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
4. Define settings for a **Variant**, which are applied if the associated target's conditions are met.

5. Set the value of `SIProtocols` to include the additional supported protocols you want to add. For example, `http;https;wsp`. Note that the value must be of type REG\_MULTI\_SZ.

**Testing steps:**

Work with your mobile operator to test this customization on their network.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Switch from SMS to MMS for long messages

10/2/2018 • 2 minutes to read • [Edit Online](#)

For networks that do support MMS and do not support segmentation of SMS messages, partners can specify an automatic switch from SMS to MMS for long messages.

**Constraints:** None

This customization supports: **per-SIM** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="ConvertLongSMSToMMS"
    Description="Use to specify an automatic switch from SMS to MMS for long messages."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Messaging/PerSimSettings/$(__ICCID)">
            <Setting Name="ConvertLongSMSToMMS" Value="" />
        </Settings>
    </Variant>
    </ImageCustomizations>
```

2. Specify an **Owner**.
3. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
4. Define settings for a **Variant**, which are applied if the associated target's conditions are met.

5. Set the value of `ConvertLongSMSToMMS` value to one of the following:

VALUE	DESCRIPTION
0 or 'False'	Disables switching from SMS to MMS for long messages. This is the default behavior.
1 or 'True'	Enables switching from SMS to MMS for long messages.

**Testing steps:**

Work with your mobile operator to test this customization on their network.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Truncated content handling for WAP push notification

10/2/2018 • 2 minutes to read • [Edit Online](#)

For networks that require non-standard handling of single-segment incoming MMS WAP Push notifications, partners can specify that MMS messages may have some of their content truncated and that they may require special handling to reconstruct truncated field values.

**Constraints:** None

This customization supports: **per-SIM** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="WapPushTechnology"
    Description="Use to specify that MMS messages may have some of their content
truncated and to indicate that the message may require special handling to reconstruct
truncated field values."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Messaging/PerSimSettings/$(__ICCID)">
            <Setting Name="WapPushTechnology" Value="" />
        </Settings>
    </Variant>
</ImageCustomizations>
```

2. Specify an `Owner`.

3. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and

SPN.

4. Define settings for a **Variant**, which are applied if the associated target's conditions are met.

5. Set `WapPushTechnology` to one of the following values:

VALUE	DESCRIPTION
1 or 0x1	Enables MMS messages to have some of their content truncated.
0 or 0x0	Disables MMS messages from being truncated.

**Testing steps:**

Work with your mobile operator to test this customization on their network.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Use insert-address-token or local raw address

10/2/2018 • 2 minutes to read • [Edit Online](#)

To meet certain mobile operator requirements, OEMs can customize the OS image to use either the insert-address-token or the local raw address for the **From** field in MMS messages.

**Constraints:** None

This customization supports: **per-SIM** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="UseInsertAddressToken"
    Description="Configure if you want to use insert-address-token or the local raw
address for
    the 'From' field in MMS messages."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Messaging/PerSimSettings/${__ICCID}">
            <Setting Name="UseInsertAddressToken" Value="" />
        </Settings>

    </Variant>
</ImageCustomizations>
```

2. Specify an **Owner**.
3. Define the **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.

4. Define the settings for a **Variant**, which are applied if the associated target's conditions are met.

5. Set the value for `UseInsertAddressToken` to one of the following values:

VALUE	DESCRIPTION
0 or 'False'	Uses the local raw address for the <b>From</b> field for MMS messages.
1 or 'True'	Uses the insert-address-token for the <b>From</b> field for MMS messages. This is the default value set by the OS.

**Testing steps:**

Work with your mobile operator to properly test this customization on their network.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Use UTF-8 for MMS messages with unspecified character encoding

10/2/2018 • 2 minutes to read • [Edit Online](#)

Some incoming MMS messages may not specify a character encoding. To properly decode MMS messages that do not specify a character encoding, OEMs can set UTF-8 to decode the message.

**Constraints:** None

This customization supports: **per-SIM** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="UseUTF8ForUnspecifiedCharset"
    Description="Use to set UTF-8 character encoding to decode incoming MMS messages
that do not have a specified
character encoding."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Messaging/PerSimSettings/$__ICCID">
            <Setting Name="UseUTF8ForUnspecifiedCharset" Value="" />
        </Settings>
    </Variant>
</ImageCustomizations>
```

2. Specify an **Owner**.
3. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and

SPN.

4. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
5. Set the value of `UseUTF8ForUnspecifiedCharset` to one of the following:

VALUE	DESCRIPTION
0 or 'False'	Disables using UTF-8 for MMS messages with unspecified character encoding. This is the default behavior.
1 or 'True'	Enables using UTF-8 for MMS messages with unspecified character encoding.

#### Testing steps:

Work with your mobile operator to properly test this customization on their network.

To verify the customization:

1. Use a device to send an MMS message without a character encoding specified to your Windows 10 Mobile device.
2. Verify that the MMS was received correctly on your Windows 10 Mobile device. The message should not contain any unrecognized characters.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# User agent profile for MMS messages

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can specify a user agent profile to use on the device for MMS messages. The user agent profile XML file details a device's hardware specifications and media capabilities so that an MMS application server (MMSC) can return supported optimized media content to the device. The user agent profile XML file is generally stored on the MMSC.

**Constraints:** None

This customization supports: **per-SIM** value

## Instructions:

The MMS-specific components for Windows 10 Mobile are included in the following code sample.

```
<!-- **** Microsoft MMS Baseline Characteristics Description ****
5/25/2012 -->
<prf:component>
  <rdf:Description rdf:ID="MMSCharacteristics">
    <rdf:type rdf:resource=
      "http://www.wapforum.org/profiles/MMS/ccpschema-20010111#MmsCharacteristics" />
    <!-- Max size must mirror max size that the mobile operator
        allows for sending. -->
    <mms:MmsMaxMessageSize>614400</mms:MmsMaxMessageSize>
    <mms:MmsMaxImageResolution>1600x1600</mms:MmsMaxImageResolution>
    <mms:MmsCcppAccept>
      <rdf:Bag>
        <!-- Image -->
        <rdf:li>image/jpeg</rdf:li>
        <rdf:li>image/gif</rdf:li>
        <rdf:li>image/bmp</rdf:li>
        <rdf:li>image/png</rdf:li>
        <rdf:li>image/tiff</rdf:li>
        <rdf:li>image/wdp</rdf:li>
        <rdf:li>image/vnd.ms-photo</rdf:li>
        <!-- Audio -->
        <rdf:li>audio/mp3</rdf:li>
        <rdf:li>audio/mpeg</rdf:li>
        <rdf:li>audio/mpeg3</rdf:li>
        <rdf:li>audio/mp4</rdf:li>
        <rdf:li>audio/wav</rdf:li>
        <rdf:li>audio/x-ms-wav</rdf:li>
        <rdf:li>audio/vnd.wave</rdf:li>
        <rdf:li>audio/3gpp</rdf:li>
        <rdf:li>audio/3gpp2</rdf:li>
        <rdf:li>audio/x-ms-wma</rdf:li>
        <rdf:li>audio/aac</rdf:li>
        <rdf:li>audio/aacp</rdf:li>
        <rdf:li>audio/vnd.dlna.adts</rdf:li>
        <rdf:li>audio/x-aac</rdf:li>
        <rdf:li>audio/x-m4a</rdf:li>
        <rdf:li>audio/x-mp3</rdf:li>
        <rdf:li>audio/x-mpeg</rdf:li>
        <rdf:li>audio/x-wav</rdf:li>
        <rdf:li>audio/amr</rdf:li>
        <rdf:li>audio/x-m4r</rdf:li>
        <!-- Video -->
        <rdf:li>video/3gpp</rdf:li>
        <rdf:li>video/3gpp2</rdf:li>
        <rdf:li>video/mp4</rdf:li>
        <rdf:li>video/x-m4v</rdf:li>
        <rdf:li>video/x-ms-wmav</rdf:li>
    </rdf:Bag>
  </rdf:Description>
</prf:component>
```

```

<rdf:li>video/x-ms-wmv</rdf:li>
<rdf:li>video/quicktime</rdf:li>
<!-- Text -->
<rdf:li>text/plain</rdf:li>
<!-- VCards -->
<rdf:li>text/vcard</rdf:li>
<rdf:li>text/x-vcard</rdf:li>
<!-- Other MIME type -->
<rdf:li>application/vnd.wap.multipart.mixed</rdf:li>
<rdf:li>application/vnd.wap.multipart.related</rdf:li>
<rdf:li>application/smil</rdf:li>
<rdf:li>application/vnd.wap.mms-message</rdf:li>
<rdf:li>application/vnd.oma.drm.message</rdf:li>
</rdf:Bag>
</mms:MmsCcppAccept>
<mms:MmsCcppAcceptCharSet>
<rdf:Bag>
<rdf:li>UTF-8</rdf:li>
<rdf:li>UTF-16</rdf:li>
<rdf:li>ISO-8859-1</rdf:li>
<rdf:li>ISO-8859-2</rdf:li>
<rdf:li>ISO-8859-3</rdf:li>
<rdf:li>ISO-8859-4</rdf:li>
<rdf:li>ISO-8859-5</rdf:li>
<rdf:li>ISO-8859-6</rdf:li>
<rdf:li>ISO-8859-7</rdf:li>
<rdf:li>ISO-8859-8</rdf:li>
<rdf:li>ISO-8859-9</rdf:li>
<rdf:li>ISO-8859-10</rdf:li>
<rdf:li>US-ASCII</rdf:li>
<rdf:li>ISO-10646-UCS-2</rdf:li>
</rdf:Bag>
</mms:MmsCcppAcceptCharSet>
<mms:MmsVersion>
<rdf:Bag>
<rdf:li>1.2</rdf:li>
</rdf:Bag>
</mms:MmsVersion>
</rdf:Description>
</prf:component>

```

There are two ways to correlate a user agent profile with a given device:

- You can take the user agent string of the device that is sent with MMS requests and use it as a hash to map to the user agent profile on the MMSC. The user agent string cannot be modified.
- Alternatively, you can directly set the URI of the user agent profile on the device.

The following steps describe how to specify a custom user agent profile XML file by using a registry setting.

1. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="MMSUAProfile"
    Description="Use to specify a user agent profile to use for MMS messages."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Messaging/PerSimSettings/$(__ICCID)">

            <!-- Replace UAProf.xml with the full URI of your user agent profile file. -->
            <Setting Name="UAProf" Value="UAProf.xml" />

            <!-- Use to specify the custom user agent property name. Set the value to either
                "x-wap-profile or "profile" -->
            <Setting Name="UAProfToken" Value="x-wap-profile" />

        </Settings>
    </Variant>
</ImageCustomizations>

```

2. Specify an `Owner`.
3. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
4. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
5. In the `UAProf`Value`, replace `UAProf.xml` value with the full URI of your user agent profile.

Optionally, you can also specify the custom user agent property name for MMS that is sent in the header by setting `UAProfToken` to either `x-wap-profile` or `profile`.

#### Testing steps:

Work with your mobile operator to test this customization on their network.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# User agent string for MMS messages

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can replace the entire user agent string for MMS.

By default, this string has the format `WindowsPhoneMMS/MicrosoftMMSVersionNumber WindowsPhoneOS/OSVersion-buildNumber OEM-deviceName`, in which the *italicized text* is replaced with the appropriate values for the phone.

**Constraints:** None

This customization supports: **per-SIM** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="MMSUASTring"
    Description="Use to replace the entire user agent string for MMS messages."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Messaging/PerSimSettings/$__ICCID">
            <Setting Name="UserAgentString" Value="" />
        </Settings>

    </Variant>
</ImageCustomizations>
```

2. Specify an `Owner`.

3. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and

SPN.

4. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
5. Set the `UserAgentString``Value` to the new user agent string for MMS in its entirely.

By default, this string has the format *WindowsPhoneMMS/MicrosoftMMSVersionNumber WindowsPhoneOS/OSVersion-buildNumber OEM-deviceName*, in which the *italicized text* is replaced with the appropriate values for the device.

**Testing:**

Work with your mobile operator to test this customization on their network.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# User alert for service indication messages

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can hide the user prompts for signal-medium messages.

By default, when a service indication message is received with a signal-medium or signal-high setting, the phone interrupts and shows the user prompt for these messages. However, partners can hide the user prompts for signal-medium messages.

**Constraints:** None

This customization supports: **per-SIM** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="HideMediumSIPopups"
    Description="Use to hide the user prompts for signal-medium service indication
messages."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Messaging/PerSimSettings/$__ICCID">
            <Setting Name="HideMediumSIPopups" Value="" />
        </Settings>
    </Variant>
</ImageCustomizations>
```

2. Specify an **Owner**.

3. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and

SPN.

4. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
5. Set the value of `HideMediumSIPopups` value to one of the following:

VALUE	DESCRIPTION
0 or 'False'	Shows the signal-medium service indication messages. This is the default behavior.
1 or 'True'	Hides the signal-medium service indication messages.

**Testing steps:**

1. Flash the build containing this customization to a device that has a SIM.
2. Send a signal-medium service indication message to the device and verify that a notification window does not appear at the top of the screen.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Video attachments in MMS

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can specify the transcoding to use for video files sent as attachments in MMS messages.

**Constraints:** None

This customization supports: **per-SIM** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="TargetVideoFormat"
    Description="Use to specify the transcoding to use for video files sent as
attachments in MMS messages."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="Messaging/PerSimSettings/${__ICCID}">
            <Setting Name="TargetVideoFormat" Value="" />
        </Settings>
    </Variant>
</ImageCustomizations>
```

2. Specify an `Owner`.
3. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
4. Define settings for a **Variant**, which are applied if the associated target's conditions are met.

5. Set `TargetVideoFormat` to one of the following values to configure the default transcoding for video files sent as attachments in MMS messages:

VALUE	DESCRIPTION
0 or 0x0	Sets the transcoding to H.264 + AAC + MP4. This is the default set by the OS.
1 or 0x1	Sets the transcoding to H.264 + AAC + 3GP.
2 or 0x2	Sets the transcoding to H.263 + AMR.NB + 3GP.
3 or 0x3	Sets the transcoding to MPEG4 + AMR.NB + 3GP.

**Testing:**

1. Flash the build containing this customization to a device.
2. Attempt to send a message with an attachment that requires the new transcoding. Verify that the message sends and that the file can be opened after it is received.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Voicemail SMS intercept

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can define a filter that intercepts an incoming SMS message and triggers visual voicemail synchronization. The filtered message does not appear in the user's conversation list.

A visual voicemail sync is triggered by an incoming SMS message if the following conditions are met:

- The message sender value starts with the string specified in the `SyncSender` setting. The length of the specified values must be greater than 3 characters but less than 75 characters.
- The body of the message starts with the string specified in the `SyncPrefix` setting. The length of the specified values must be greater than 3 characters but less than 75 characters.
- Visual voicemail is configured and enabled. For more information, see [Visual voicemail](#).

## Constraints: Atomic

This customization supports: **per-SIM** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="VoicemailsSMSIntercept"
    Description="Use to define a filter that intercepts an incoming SMS message and
triggers visual voicemail synchronization."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Messaging/GlobalSettings/VoicemailIntercept">
            <Setting Name="SyncSender" Value="" />
            <Setting Name="SyncPrefix" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Specify a value for `SyncSender` that is greater than 3 characters but less than 75 characters in length.

For networks that support it, this value can be a short code of the mailbox server that sends a standard SMS notification.

4. Specify a value for `SyncPrefix` that is greater than 3 characters but less than 75 characters in length.

For networks that support it, this value can be the keyword for the SMS notification.

## Note

The `SyncSender` and `SyncPrefix` values vary for each mobile operator, so OEMs must work with their mobile operators to obtain the correct or required values.

Make sure that the correct visual voicemail settings for the mobile operator are also set. For more information, see [Visual voicemail](#).

**Testing steps:**

1. Flash the build containing this customization to a phone that has a UICC.
2. Successfully testing this customization requires the correct values, so work with your mobile operator partner to test this customization on their network.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Customizations for SIM settings

10/2/2018 • 2 minutes to read • [Edit Online](#)

Describes the customizations for SIM settings.

## In this section

TOPIC	DESCRIPTION
<a href="#">Add a suffix to the mobile operator name</a>	To meet branding requirements for some mobile operators, OEMs can add a suffix to the network name that is displayed on the phone. For example, from ABC to ABC 3G when under 3G coverage.
<a href="#">Additional Internet APN settings</a>	OEMs can hide both the add internet apn button and the IP type listbox in the internet APN settings screen.
<a href="#">Change SIM to SIM/UIM</a>	Partners can change the string "SIM" to "SIM/UIM" in the device UI.
<a href="#">Change the default SIM name to match the SPN or operator name</a>	Partners can change the default name read from the SIM to define the SPN for SIM cards that do not contain this information or to generate the default friendly name for the SIM.
<a href="#">Configure C+G dual SIM settings</a>	Partners can configure the settings for C+G dual SIM phones.
<a href="#">Hide the SIM security settings option</a>	OEMs can hide the SIM security settings option. By default, this is visible when you go to the Settings > applications > phone screen.
<a href="#">Remove the trailing MSISDN digits on a SIM card</a>	OEMs can remove the trailing MSISDN digits from the service provider name (SPN) in the device UI.
<a href="#">Settings for IMS services</a>	OEMs can configure the default settings and toggle for IMS services to meet mobile operator requirements.
<a href="#">View Internet APN</a>	For mobile operators that require it, OEMs can show the View Internet APN button in the Cellular & SIM settings page for users that have a data plan. When data is off, the button is disabled. By default, the button is hidden.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Add a suffix to the mobile operator name

10/2/2018 • 2 minutes to read • [Edit Online](#)

To meet branding requirements for some mobile operators, OEMs can add a suffix to the network name that is displayed on the phone. For example, from **ABC** to **ABC 3G** when under 3G coverage. This feature can be applied for any radio access technology (RAT).

For TD-SCDMA RAT, a 3G suffix is always appended by default, but partners can also customize this the same way as with any other RAT.

**Constraints:** None

This customization supports: **per-IMSI** value, **per-device** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="NetworkSuffix"
    Description="Use to add a suffix to the network name that is displayed on the
device."
    Owner=""
    OwnerType="OEM">

<!-- Use for the per-IMSI case

&lt;!-- Define the Targets --&gt;
&lt;Targets&gt;
    &lt;Target Id=""&gt;
        &lt;TargetState&gt;
            &lt;Condition Name="" Value="" /&gt;
            &lt;Condition Name="" Value="" /&gt;
        &lt;/TargetState&gt;
    &lt;/Target&gt;
&lt;/Targets&gt;

&lt;Static&gt;
    &lt;Settings Path="Multivariant"&gt;
        &lt;Setting Name="Enable" Value="1" /&gt;
    &lt;/Settings&gt;
    &lt;Settings Path="AutoDataConfig"&gt;
        &lt;Setting Name="Enable" Value="0" /&gt;
    &lt;/Settings&gt;
&lt;/Static&gt;

&lt;!-- Specify the Variant --&gt;
&lt;Variant Name=""&gt;
    &lt;TargetRefs&gt;
        &lt;TargetRef Id="" /&gt;
    &lt;/TargetRefs&gt;

    &lt;Settings Path="CellCore/PerIMSI/$__IMSI/General"&gt;
        &lt;Setting Name="NetworkSuffix/$(SYSTEMTYPE)" Value="" /&gt;
    &lt;/Settings&gt;

&lt;/Variant&gt;

--&gt;

<!-- Use for the per-device case

&lt;Static&gt;
    &lt;Settings Path="CellCore/PerDevice/General"&gt;
        &lt;Setting Name="NetworkSuffix/$(SYSTEMTYPE)" Value="" /&gt;
    &lt;/Settings&gt;
&lt;/Static&gt;

--&gt;

&lt;/ImageCustomizations&gt;
</pre>

```

2. Specify an `Owner`.

3. Determine if you need to use the **per-IMSI** or **per-device** setting.

For the **per-IMSI** case:

- Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
- Define settings for a **Variant**, which are applied if the associated target's conditions are met.

4. In the setting name, set `$(SYSTEMTYPE)` to the network type that you want to append the network name to, and set the value for that network type as shown in the following table:

<code>\$(SYSTEMTYPE)</code>	VALUE	DESCRIPTION
4	2G	Represents RIL_SYSTEMTYPE_GSM (GSM connection).
8	3G	Represents RIL_SYSTEMTYPE_UMTS (UMTS connection).
16	LTE	Represents RIL_SYSTEMTYPE_LTE (LTE connection).
32	3G	<p>Represents RIL_SYSTEMTYPE_TDSCDMA (TD-SCDMA connection).</p> <p>Partners do not need to set this registry value.</p> <p>By default, this registry setting is set to " 3G".</p>

For example, if you would like the mobile operator name (ABC) and the suffix (3G) to appear as \*\*ABC 3G\*\* (there is a space between the network name and the connection type) rather than \*\*ABC3G\*\* (no space between the network name and connection type), you need to include a space when setting the values.

### Testing steps:

1. Flash a build containing this customization to a device with a SIM.
2. In the Start screen, verify that the phone tile shows the mobile operator name followed one of the suffixes that you set.  
The suffix will depend on the network.
3. Go to the **Settings** screen and scroll down until you see **Cellular & SIM**. Verify that the mobile operator name is followed by the correct suffix. This should match what you see in the phone tile in the Start screen.
4. Tap **Cellular & SIM** and verify that the **Active network** shows the mobile operator name and correct suffix.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Additional Internet APN settings

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can hide both the **add internet apn** button and the **IP type** listbox in the **internet APN** settings screen.

If it is required by the mobile operator OEMs can hide the **add internet apn** button, which enables the user to manually add and configure a data connection for a network. OEMs can also hide the **IP type** listbox in the **internet APN** settings screen.

## Limitations and restrictions:

- Partners must not provide an alternate user interface for adding or editing APN values.

## Constraints:

None  
This customization supports: **per-IMSI** value, **per-device** value

## Instructions:

- Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="InternetAPNSettings"
    Description="Use to hide or show the 'add internet apn' button in the SIM settings
screen,
                                and hide or show the 'IP type' setting in the internet APN settings
screen."
    Owner=""
    OwnerType="OEM">
```

```

<!-- Use for the per-IMSI case

<!-- Define the Targets -->
<Targets>
    <Target Id="">
        <TargetState>
            <Condition Name="" Value="" />
            <Condition Name="" Value="" />
        </TargetState>
    </Target>
</Targets>

<Static>
    <Settings Path="Multivariant">
        <Setting Name="Enable" Value="1" />
    </Settings>
    <Settings Path="AutoDataConfig">
        <Setting Name="Enable" Value="0" />
    </Settings>
</Static>

<!-- Specify the Variant -->
<Variant Name="">
    <TargetRefs>
        <TargetRef Id="" />
    </TargetRefs>

    <Settings Path="CellCore/PerIMSI/$(__IMSI)/CellUX">
        <Setting Name="HideAPN" Value="" />
        <Setting Name="HideAPNIPType" Value="" />
        <Setting Name="APNIPTypeIfHidden" Value="" />
    </Settings>
</Variant>

-->

<!-- Use for the per-device case

<Static>
    <Settings Path="CellCore/PerDevice/CellUX">
        <Setting Name="HideAPN" Value="" />
        <Setting Name="HideAPNIPType" Value="" />
        <Setting Name="APNIPTypeIfHidden" Value="" />
    </Settings>
</Static>

-->

</ImageCustomizations>
```

```

1. Specify an `Owner`.
2. Determine if you need to use the **per-IMSI** or **per-device** setting.

For the **per-IMSI** case:

- a. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
  - b. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
3. Set the value for `HideAPN` to one of the following:

| VALUE      | DESCRIPTION   |
|------------|---|
| 0 or 'No'  | Shows the <b>add internet APN</b> button in the <b>SIM</b> settings screen. |
| 1 or 'Yes' | Hides the <b>add internet APN</b> button in the <b>SIM</b> settings screen. |

4. Set the value for `HideAPNIPType` to one of the following:

| VALUE      | DESCRIPTION  |
|------------|--|
| 0 or 'No'  | Shows the <b>IP type</b> listbox in the <b>internet APN</b> settings screen. |
| 1 or 'Yes' | Hides the <b>IP type</b> listbox in the <b>internet APN</b> settings screen. |

5. To change the default IP type shown in the **IP type** listbox: Set the value for `APNIPTypeIfHidden` to one of the following:

| VALUE             | DESCRIPTION                              |
|-------------------|--|
| 0 or 'IPV4'       | Sets the default IP type to IPv4.        |
| 1 or 'IPV6'       | Sets the default IP type to IPv6.        |
| 2 or 'IPV4v6'     | Sets the default IP type to IPv4v6.      |
| 3 or 'IPV4v6XLAT' | Sets the default IP type to IPv4v6 XLAT. |

## Testing:

1. Flash the build containing this customization to a device.
2. Go to the **cellular & SIM** screen in **Settings**.
3. Verify that the **add internet apn** button is no longer visible if configured to be hidden.
4. Tap the **add internet apn** button. Depending on the setting, verify that:
  - a. The **IP type** setting either shows a dropdown listbox with **IPv4**, **IPv6**, **IPv4v6**, or **IPv4v6 464XLAT**.  
Or,
  - b. The **IP type** setting is hidden.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Change SIM to SIM/UIM

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can change the string "SIM" to "SIM/UIM" in the device UI.

Enabling this customization changes all "SIM" strings to "SIM/UIM".

**Constraints:** None

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="SIMToSIMUIM"
    Description="Use to replace the 'SIM' strings in the device UI to 'SIM/UIM' to
    accommodate scenarios such as Dual Mode cards of
    SIM cards on the device."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="CellCore/PerDevice/UIX">
            <!-- Set the value to 0 or "SIM" (to use the default SIM string), or set to 1 or "UIM" (to use the
            alternate SIM/UIM string) -->
            <Setting Name="SIMToSIMUIM" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set the value of `SIMToSIMUIM` to one of the following:

VALUE	DESCRIPTION
0 or 'SIM'	Uses the default string "SIM".
1 or 'UIM'	Uses the alternate string "SIM/UIM".

## Testing steps:

1. Flash a build containing this customization to a device without a SIM.
2. During the initial device setup, verify that you see the following error:

### **SIM/UIM error**

**The SIM/UIM card is missing or invalid. You can still make emergency calls if your mobile operator supports it.**

3. In the Start screen, verify that the phone tile shows **No SIM/UIM**.

All other "SIM" strings on the device should now show "SIM/UIM".

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Change the default SIM name to match the SPN or operator name

10/2/2018 • 2 minutes to read • [Edit Online](#)

By default, the OS displays **SIM 1** or **SIM 2** as the default friendly name for the SIM in slot 1 or slot 2 if the service provider name (SPN) or mobile operator name has not been set. Partners can use this customization to change the default name read from the SIM to define the SPN for SIM cards that do not contain this information or to generate the default friendly name for the SIM.

The OS uses the default value as the display name for the SIM or SPN in the Start screen and other parts of the UI including the SIM settings screen. For dual SIM phones that contain SIMs from the same mobile operator, the names that appear in the UI may be similar.

**Constraints:** None

This customization supports: **per-IMSI** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="MultivariantProvisionedSPN"
    Description="Use to define the SPN for SIM cards that don't contain this
information or use to
                                generate the default friendly name for the SIM."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="CellCore/PerIMSI/$(__IMSI)/General/Critical">
            <Setting Name="MultivariantProvisionedSPN" Value="" />
        </Settings>

    </Variant>
</ImageCustomizations>

```

2. Specify an `Owner`.

3. Set the `MultivariantProvisionedSPN` value to the name of the SPN or mobile operator.

The following table shows the scenarios supported by this customization:

#### NOTE

In the **Default SIM name** column:

- The " " in `MultivariantProvisionedSPN` "1234 means that there is a space between the mobile operator name or SPN and the last 4 digits of the MSISDN.
- `MultivariantProvisionedSPN` means the value that you set for the `MultivariantProvisionedSPN` setting.
- `SIM 1 or SIM 2` is the default friendly name for the SIM in slot 1 or slot 2.

MULTIVARIANT SETTING SET?	SPN PROVISIONED?	MSISDN (LAST 4 DIGITS: 1234, FOR EXAMPLE) PROVISIONED?	DEFAULT SIM NAME
------------------------------	------------------	--	------------------

MULTIVARIANT SETTING SET?	SPN PROVISIONED?	MSISDN (LAST 4 DIGITS: 1234, FOR EXAMPLE) PROVISIONED?	DEFAULT SIM NAME
Yes	Yes	Yes	<i>MultivariantProvisionedSPN1234 or MultivariantProvisionedSPN"1234</i>
Yes	No	No	<i>MultivariantProvisionedSPN</i> (up to 16 characters)
Yes	Yes	No	<i>MultivariantProvisionedSPN</i> (up to 16 characters)
Yes	No	Yes	<i>MultivariantProvisionedSPN1234 or MultivariantProvisionedSPN"1234</i>
No	Yes	Yes	If SPN string >= 12: <i>SPN1234</i> If SPN string < 12: <i>SPN"1234</i>
No	No	No	<i>SIM 1 or SIM 2</i>
No	Yes	No	SPN (up to 16 characters)
No	No	Yes	<i>SIM 1 or SIM 2</i>

### Testing steps:

1. Flash a build containing this customization to a phone that supports either a single or dual SIM.
2. Boot the phone and verify that the displayed friendly name for the SIM matches the SPN name or the value set for `MultivariantProvisionedSPN`.

If there are two SIMs, verify that the displayed friendly names appear as expected.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Configure C+G dual SIM settings

10/2/2018 • 5 minutes to read • [Edit Online](#)

Partners can configure the settings for C+G dual SIM phones. The first slot is for CDMA (C) and the second slot is for GSM (G).

**Constraints:** None

This customization supports: **per-IMSI** value, **per-device** value.

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="CGSettings"
    Description="Use to configure settings for C+G dual SIM phones."
    Owner=""
    OwnerType="OEM">

    <!-- Use for the per-IMSI case.

        <!-- Define the Targets. -->
        <Targets>
            <Target Id="">
                <TargetState>
                    <Condition Name="" Value="" />
                    <Condition Name="" Value="" />
                </TargetState>
            </Target>
        </Targets>

        <Static>
            <Settings Path="Multivariant">
                <Setting Name="Enable" Value="1" />
            </Settings>
            <Settings Path="AutoDataConfig">
                <Setting Name="Enable" Value="0" />
            </Settings>
        </Static>

        <!-- Specify the Variant. -->
        <Variant Name="">
            <TargetRefs>
                <TargetRef Id="" />
            </TargetRefs>

            <Settings Path="CellCore/PerIMSI/$(__IMSI)/CellUX">
                <Setting Name="ShowManualAvoidance" Value="" />
            </Settings>

            <Settings Path="CellCore/PerIMSI/$(__IMSI)/General">
                <Setting Name="CardLock" Value="" />
                <Setting Name="CardAllowList" Value="" />
                <Setting Name="CardBlockList" Value="" />
                <Setting Name="SuggestDataRoamingARD" Value="" />
            </Settings>

            <Settings Path="DeviceInfo/Variant">
                <Setting Name="RoamingSupportPhoneNumber" Value="" />
            </Settings>
        </Variant>
    </-- Use for the per-device case. -->
</ImageCustomizations>
```

```

    </Variant>

-->

<!-- Use for the per-device case -->
<Static>

    <Settings Path="CellCore/PerDevice/CellUX">
        <Setting Name="ShowManualAvoidance" Value="" />
    </Settings>

    <Settings Path="CellCore/PerDevice/CGDual">
        <Setting Name="RestrictToGlobalMode" Value="" />
    </Settings>

    <Settings Path="CellCore/PerDevice/UIX">
        <Setting Name="SIM1ToUIM1" Value="" />
    </Settings>

    <Settings Path="CellCore/PerDevice/General">
        <Setting Name="CardLock" Value="" />
        <Setting Name="CardAllowList" Value="" />
        <Setting Name="CardBlockList" Value="" />

        <Setting Name="DefaultSlotAffinity" Value="" />
        <Setting Name="Slot2DisableAppsList" Value="" />
        <Setting Name="SuggestGlobalModeARD" Value="" />
        <Setting Name="SuggestGlobalModeTimeout" Value="" />
        <Setting Name="SuppressDePersoUI" Value="" />
        <Setting Name="SuggestDataRoamingARD" Value="" />
    </Settings>

    <Settings Path="AutomaticTime">
        <Setting Name="PreferredSlot" Value="" />
    </Settings>

</Static>

</ImageCustomizations>

```

2. Specify an **Owner**.

3. To show the **Switch to next network manually** button on the **Settings > Cellular & SIM** page, set **ShowManualAvoidance** to one of the following values:

VALUE	DESCRIPTION
0 or <input type="checkbox"/> No	Does not show the <b>Switch to next network manually</b> button.
1 or <input checked="" type="checkbox"/> Yes	Shows the <b>Switch to next network manually</b> button.

#### NOTE

This setting supports per-IMSI or per-device values. Determine which one you would like to set. By default, this setting is off or missing.

4. To configure card lock support, use and set the following settings:

- To enforce either the card allow list or both the card allow and block lists, set **CardLock** to one of the following values:

VALUE	DESCRIPTION
0 or <code>PersoLockType_AllowAndBlockList</code>	Enforces both the card allow list and the card block list.
1 or <code>PersoLockType_AllowListOnly</code>	Enforces only the card allow list.

Set the values for the `CardAllowList` and `CardBlockList` settings to configure the allow list and block list, respectively.

- b. To configure the list of SIM cards allowed in the first slot, set the value for `CardAllowList` to a comma separated MCC:MNC list. You can also use wild cards, represented by an asterisk (\*), to accept any value. For example, you can set the value to `310:410,311:*,404:012,310:70`.
- c. To configure the list of SIM cards that are not allowed in the first slot, set the value for `CardBlockList` to a comma separated MCC:MNC list. You can also use wild cards, represented by an asterisk (\*), to accept any value. For example, you can set the value to `310:410,311:*,404:012,310:70`.

#### NOTE

These settings support per-IMSI or per-device values. Determine which one you would like to set. By default, this setting is off or missing.

5. To specify the OEM or mobile operator's roaming support contact phone number, set the optional **RoamingSupportPhoneNumber** setting to the phone number you want to use. This string appears in the **About** settings screen.

This setting is part of the `DeviceInfo/Variant` settings group. For more information about the other device info settings, see [Phone metadata in DeviceTargetingInfo](#).

6. To configure the **Mode selection** in the **Cellular & SIM** settings page, set `RestrictToGlobalMode` to one of the following values:

VALUE	DESCRIPTION
0 or <code>RestrictToGlobalMode_Disabled</code>	The phone is not restricted to global mode.
1 or <code>RestrictToGlobalMode_Home</code>	When a slot is registered at home and supports global mode, the mode selection is restricted to global mode.
2 or <code>RestrictToGlobalMode_Always</code>	If a slot supports global mode and this value is selected, the mode selection is restricted to global mode.

This setting is required for phones shipping on the China Telecom network. By default, this setting is off or missing.

When the device registration changes, if the value for this setting is set, the OS changes the preferred system type to the default preferred system type for world mode. If the phone is not camped on any network, the OS assumes the phone is on the home network and changes the network registration preference to default mode.

7. To show **UIM1** as an alternate string instead of **SIM1** for the first SIM, set `SIM1ToUIM1` to one of the following values:

VALUE	DESCRIPTION
0 or <code>No</code>	Keeps the <b>SIM1</b> strings in the UI for dual SIM phones.
1 or <code>Yes</code>	Changes the <b>SIM1</b> strings in the UI to <b>UIM1</b> for dual SIM phones.

By default, this setting is off or missing.

8. To set the data connection preference, set `DefaultSlotAffinity` to one of the following values:

VALUE	DESCRIPTION
0 or <code>SlotAffinityForInternetData_Automatic</code>	The data connection preference is automatically set. When set, the OS shows the <b>Two SIMs?</b> page when trying to identify the network.
1 or <code>SlotAffinityForInternetData_Slot0</code>	Sets the data connection preference to Slot 0 and the data connection cannot be edited.
2 or <code>SlotAffinityForInternetData_Slot1</code>	Sets the data connection preference to Slot 1 and the data connection cannot be edited.

9. To disable a list of specified apps from Slot 2, set `Slot2DisableAppsList` to a comma separated value. For example, `4,6`.

10. To suggest global mode when the phone is not registered on other modes in the network, set `SuggestGlobalModeARD` to one of the following values:

VALUE	DESCRIPTION
0 or <code>SuggestGlobalModeARD_Disable</code>	No suggestion is made.
1 or <code>SuggestGlobalModeARD_Enable</code>	Global mode is suggested.

11. To specify the number of seconds to wait for the network registration before suggesting global mode, set `SuggestGlobalModeTimeout` to a value between 1 and 600, inclusive. For example, to set the timeout to 60 seconds, set the value to 60 (decimal) or `0x3C` (hexadecimal).

12. To suppress the perso unlock UI, set `SuppressDePersoUI` to one of the following values:

VALUE	DESCRIPTION
0 or <code>Disable</code>	Shows the perso unlock UI.
1 or <code>Enable</code>	Suppresses the perso unlock UI.

13. To show the data roaming suggestion dialog when roaming and the data roaming setting is set to no roaming, set `SuggestDataRoamingARD` to one of the following values:

VALUE	DESCRIPTION
0 or <code>SuggestDataRoamingARD_Disable</code>	No suggestion is made.

VALUE	DESCRIPTION
1 or <code>SuggestDataRoamingARD_Enable</code>	Data roaming suggestion is made.

14. To specify which UICC slot will be used for NITZ handling, set `PreferredSlot` to one of the following values:

VALUE	DESCRIPTION
0 or <code>Slot 0</code>	Uses the UICC in Slot 0 for NITZ handling.
1 or <code>Slot 1</code>	Uses the UICC in Slot 1 for NITZ handling.

#### Testing:

Work with your mobile operator to determine the setting requirements for the network.

1. Flash the build containing this customization to a C+G dual SIM phone.
2. Depending on the values you specified for the C+G settings, verify that the behavior matches the setting value description.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Hide the SIM security settings option

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can hide the **SIM security** settings option.

By default, the is visible when you go to the **Settings > applications > phone** screen. To meet certain mobile operator requirements or to provide a better user experience (including scenarios where a device contains a brand new SIM that doesn't require a security PIN), OEMs can hide this settings option.

**Constraints:** FirstVariationOnly

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="HideSIMSecurityUI"
    Description="Use to hide the 'SIM security' settings option from the Phone settings
screen."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Phone/PhoneSettings">
            <Setting Name="HideSIMSecurityUI" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.
3. Set the `HideSIMSecurityUI` setting to one of the following values:

VALUE	DESCRIPTION
0 or 'False'	Show the <b>SIM security</b> settings option in the <b>Settings &gt; applications &gt; phone</b> screen. This is the default OS behavior.
1 or 'True'	Hide the <b>SIM security</b> settings option in the <b>Settings &gt; applications &gt; phone</b> screen.

## Testing steps:

Work with your mobile operator partner to test this customization on their network.

1. Flash the build containing this customization to a device that has a brand new SIM that doesn't require a security PIN.
2. Set up the device.

3. If you set `HideSIMSecurityUI` to 1 or 'True', go to the **Phone** settings screen and verify that the **SIM security** settings option is not visible.
4. If you set `HideSIMSecurityUI` to 0 or 'False' or you did not change the default OS value, go to the **Phone** settings screen and verify that the **SIM security** settings option is visible.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Remove the trailing MSISDN digits on a SIM card

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can remove the trailing MSISDN digits from the service provider name (SPN) in the device UI.

By default, the OS appends the trailing MSISDN digits to the service provider name (SPN) in the device UI, including on the device and messaging apps. If required by mobile operators, OEMs can use the

`SimNameWithoutMSISDNEabled` setting to remove the trailing MSISDN digits. However, you must use this setting together with `MultivariantProvisionedSPN` to suppress the MSISDN digits. For more information about how to use `MultivariantProvisionedSPN`, see [Change the default SIM name to match the SPN or operator name](#).

**Constraints:** None

This customization supports: **per-IMSI** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="SimNameWithoutMSISDNEabled"
    Description="Use to suppress the trailing MSISDN digits that are appended to the
    SPN in the device UI."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="CellCore/PerIMSI/$(__IMSI)/General/Critical">
            <Setting Name="MultivariantProvisionedSPN" Value="" />
            <Setting Name="SimNameWithoutMSISDNEabled" Value="" />
        </Settings>

    </Variant>
</ImageCustomizations>
```

2. Specify an `Owner`.
3. Set the `MultivariantProvisionedSPN` value to the name of the SPN or mobile operator. For more information, see [Change the default SIM name to match the SPN or operator name](#).
4. Set `SimNameWithoutMSISDNEabled` to one of the following values:

VALUE	DESCRIPTION
0 or 'No'	<p>Keeps the trailing MSISDN digits.</p> <p>This is the default OS behavior.</p>
1 or 'Yes'	Removes the trailing MSISDN digits.

#### Testing steps:

1. Flash a build containing this customization to a device that supports either a single or dual SIM.
2. Boot the device and verify the following:
  - The displayed friendly name for the SIM matches the SPN name or the value set for `MultivariantProvisionedSPN`. If there are two SIMs, verify that the displayed friendly names appear as expected for that SIM.
  - If you set `SimNameWithoutMSISDNEabled` = 1, the trailing MSISDN digits should not appear. If there are two SIMs and both have `SimNameWithoutMSISDNEabled` = 1, both SIMs should not show the MSISDN digits in the UI.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Settings for IMS services

10/2/2018 • 7 minutes to read • [Edit Online](#)

OEMs can configure the default settings and toggle for IMS services to meet mobile operator requirements. Users can later manually change the default values for these settings if they choose to do so.

**Constraints:** None

This customization supports: **per-IMSI** value, **per-device** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="IMSSettings"
    Description="Use to configure the toggles and other settings for IMS services."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>

        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>

        <Settings Path="Localization/MUI">
            <!-- Use to add your base MUI DLL file -->
            <Asset Name="BaseDll" Source="" />

            <!-- Use to specify the language MUI packages (*.dll.mui) for the languages you are supporting
            and have localized strings for -->
            <Asset Name="LanguageDll/${langid}" Source="" />
            <Asset Name="LanguageDll/${langid}" Source="" />
            <Asset Name="LanguageDll/${langid}" Source="" />
            <!-- Add as many as you need -->
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <Settings Path="CellCore/PerIMSI/$_IMSI/CellUX/Critical">
            <Setting Name="ShowVoLTEToggle" Value="" />
        </Settings>
    </Variant>
</ImageCustomizations>
```

```

<Setting Name="SwitchIMS" Value="" />
<Setting Name="SwitchSMSOverIMS" Value="" />
<Setting Name="SwitchVoiceOverIMS" Value="" />
<Setting Name="SwitchVideoOverIMS" Value="" />
<Setting Name="SwitchXCAP" Value="" />
<Setting Name="VoLTEToggleDescription" Value="" />
<Setting Name="VoLTEToggleTitle" Value="" />

<!-- Settings for IMS roaming -->
<Setting Name="ShowVoLTERoaming" Value="" />
<Setting Name="VoLTESectionTitle" Value="" />
<Setting Name="VoLTERoamingTitle" Value="" />
<Setting Name="VoLTERoamingOnDescription" Value="" />
<Setting Name="VoLTERoamingOffDescription" Value="" />

<!-- Settings during active VoLTE calls -->
<Setting Name="VoLTESettingDisableDuringCall" Value="" />
<Setting Name="WFCSettingDisableDuringCall" Value="" />
<Setting Name="VoLTEToggleSettingDisableDuringCall" Value="" />
<Setting Name="VoLTERoamingSettingDisableDuringCall" Value="" />

</Settings>

<Settings Path="CellCore/PerDevice/CellUX/Critical">
    <!-- Use to hide or show the VoLTE toggle in the Settings > Cellular+SIM > SIM settings screen.
        <Setting Name="ShowVoLTEToggle" Value="" />
    </Settings>

</Variant>

</ImageCustomizations>

```

2. Specify an `Owner`.

3. Add the resource-only .dll file and the language MUI packages (\*.dll.mui) for the languages you are supporting. To do this, follow these steps:

- Add the resource-only .dll that contains the custom display string by setting the `BaseDll` asset to point to the location of your base MUI DLL file. For example: `C:\Path\DisplayStrings.dll`.
- Add the language MUI packages (\*.dll.mui) for all the languages you are supporting and have localized strings for. To do this:
  - Set the asset's `Name` to `LanguageDll/ $(langid)` where `$(langid)` corresponds to the language. For example: `LanguageDll/en-US`.
  - Set the asset's `Source` to the location of the .dll.mui file for that language. For example: `C:\Path\en-us\DisplayStrings.dll.mui`.
  - Repeat the previous steps for the other languages.

The following example shows the customization answer file entries for en-US, fr-CA, and es-MX languages:

```

<Asset Name="LanguageDll/en-US" Source="C:\Path\en-us\DisplayStrings.dll.mui" />
<Asset Name="LanguageDll/fr-CA" Source="C:\Path\fr-CA\DisplayStrings.dll.mui" />
<Asset Name="LanguageDll/es-MX" Source="C:\Path\es-MX\DisplayStrings.dll.mui" />

```

For more information, see [Create a resource-only .dll for localized strings](#).

4. Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.

5. Define settings for a **Variant**, which are applied if the associated target's conditions are met.

6. Set the value for the following settings:

- a. Set the value for `ShowVoLTEToggle` to one of the following to hide or show the VoLTE toggle.

VALUE	DESCRIPTION
0 or 'No'	Hides the VoLTE toggle in the <b>Settings &gt; Cellular+SIM &gt; SIM</b> screen.
1 or 'Yes'	Shows the VoLTE toggle in the <b>Settings &gt; Cellular+SIM &gt; SIM</b> screen.

- b. Set the value for `SwitchIMS` to one of the following to switch IMS off or on with a toggle.

VALUE	DESCRIPTION
0 or 'No'	The IMS service is not configured with the toggle.
1 or 'Yes'	The IMS service will be switched on or off with the toggle.

- c. Set the value for `SwitchSMSOverIMS` to one of the following to switch SMS over IMS on or off when VoLTE is enabled.

VALUE	DESCRIPTION
0 or 'No'	SMS over IMS is not configured with the toggle.
1 or 'Yes'	SMS over IMS is switched on or off with the toggle.

- d. Set the value for `SwitchVoiceOverIMS` to one of the following to switch voice over IMS on or off when VoLTE is enabled.

VALUE	DESCRIPTION
0 or 'No'	Voice over IMS is not configured with the toggle.
1 or 'Yes'	Voice over IMS is switched on or off with the toggle.

- e. Set the value for `SwitchVideoOverIMS` to one of the following to switch video over IMS on or off when VoLTE is enabled.

VALUE	DESCRIPTION
0 or 'No'	Video over IMS is not configured with the toggle.

VALUE	DESCRIPTION
1 or 'Yes'	Video over IMS is switched on or off with the toggle.

- f. Set the value for `SwitchXCAP` to one of the following to switch the XML Configuration Access Protocol (XCAP) on or off when VoLTE is enabled.

VALUE	DESCRIPTION
0 or 'No'	XCAP is not configured with the toggle.
1 or 'Yes'	XCAP is switched on or off with the toggle.

- g. a. To customize the VoLTE toggle description, set the `VoLTEToggleDescription` value to the name of the resource-only .dll file and specify the string offset. For example: `@DisplayStrings.dll,-101`.

Replace `DisplayStrings.dll` with the name of your .dll file and replace `Offset` with the correct offset for the localized string.

- b. To customize the VoLTE toggle label, set the `VoLTEToggleTitle` value to the name of the resource-only .dll file and specify the string offset. For example: `@DisplayStrings.dll,-102`.

Replace `DisplayStrings.dll` with the name of your .dll file and replace `Offset` with the correct offset for the localized string.

7. Set the value for the following IMS roaming settings:

- a. To show the IMS roaming control in the cellular settings page, set `ShowVoLTERoaming` to one of the following values:

VALUE	DESCRIPTION
0 or 'No'	Hides the VoLTE roaming control in the <b>Settings</b> > <b>Cellular+SIM</b> > <b>SIM</b> screen.
1 or 'Yes'	Shows the VoLTE roaming control in the <b>Settings</b> > <b>Cellular+SIM</b> > <b>SIM</b> screen.

- b. To customize the section title for the IMS settings, specify a string as the value for `VoLTESectionTitle`. The string must not be longer than 127 characters.
- c. To customize the description string for the IMS roaming control, specify a string as the value for `VoLTERoamingTitle`. The string must not be longer than 127 characters.
- d. To customize the description that appears under the IMS roaming control when IMS roaming is turned on, specify a string as the value for `VoLTERoamingOnDescription`. The string must not be longer than 127 characters.
- e. To customize the description that appears under the IMS roaming control when IMS roaming is turned off, specify a string as the value for `VoLTERoamingOffDescription`. The string must not be longer than 127 characters.

8. You can customize the settings related to active VoLTE calls by configuring these settings:

- To specify whether to grey out VoLTE-related settings during an active VoLTE call, set

`VoLTESettingDisableDuringCall` to one of the following values:

VALUE	DESCRIPTION
0 or 'No'	VoLTE-related settings are not greyed out.
1 or 'Yes'	VoLTE-related settings are greyed out.

- To specify whether to grey out Wi-Fi calling settings during an active VoLTE call, set the value for

`WFCSettingDisableDuringCall` to one of the following values:

VALUE	DESCRIPTION
0 or 'No'	Wi-Fi calling settings are not greyed out.
1 or 'Yes'	Wi-Fi calling settings are greyed out.

- To specify whether to grey out the VoLTE toggle during an active VoLTE call, set the value for

`VoLTEToggleSettingDisableDuringCall` to one of the following values:

VALUE	DESCRIPTION
0 or 'No'	The VoLTE toggle is not greyed out.
1 or 'Yes'	The VoLTE toggle is greyed out.

- To specify whether to grey out the VoLTE roaming settings during an active VoLTE call, set the value for

`VoLTERoamingSettingDisableDuringCall` to one of the following values:

VALUE	DESCRIPTION
0 or 'No'	The VoLTE roaming settings are not greyed out.
1 or 'Yes'	The VoLTE roaming settings are greyed out.

## 9. Important

Beginning with Windows Phone 8.1 GDR1, the `ShowVoLTEToggle` setting under the CellCore/PerDevice/CellUX/Critical settings path must be set. This ensures that OMA-DM VoLTE features for certain mobile operators are functioning properly. Once you have made this change to your answer file, the `ShowVoLTEToggle` setting in the CellCore/PerIMSI/\$\_IMSI)/CellUX/Critical settings path will be ignored. You may remove or leave the per-IMSI setting in your answer file.

Set the value for `ShowVoLTEToggle` to one of the following to hide or show the VoLTE toggle.

VALUE	DESCRIPTION
0 or 'No'	Hides the VoLTE toggle in the <b>Settings &gt; Cellular+SIM &gt; SIM</b> screen.
1 or 'Yes'	Shows the VoLTE toggle in the <b>Settings &gt; Cellular+SIM &gt; SIM</b> screen.

### Testing:

1. Flash the build containing this customization to a phone.
2. Go to the **Settings > Cellular & SIM** settings screen.
3. Verify that the correct settings and toggle values are showing up depending on the values you specified for `ShowVoLTEToggle` , `SwitchIMS` , `SwitchSMSOverIMS` , and `SwitchVoiceOverIMS` .
4. If you customized the VoLTE toggle label and description, verify that the correct localized strings are showing up based on the phone language.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# View Internet APN

10/2/2018 • 2 minutes to read • [Edit Online](#)

For mobile operators that require it, OEMs can show the **View Internet APN** button in the **Cellular & SIM** settings page for users that have a data plan. When data is off, the button is disabled. By default, the button is hidden.

For dual SIM devices, the button is visible depending on the multivariant SIM settings. For example, if the data plan is on SIM1 and the setting is configured to hide the button, the **View Internet APN** button will be hidden. If the user switches to SIM2 and the setting is configured to show the button, the user will see the **View Internet APN** button.

**Constraints:** None

This customization supports: **per-IMSI** value, **per-device** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="ViewInternetAPN"
    Description="Use to show the 'View Internet APN' button in the cellular+SIM
    settings screen."
    Owner=""
    OwnerType="OEM">

<!-- Use for the per-IMSI case

    &lt;!-- Define the Targets --&gt;
    &lt;Targets&gt;
        &lt;Target Id=""&gt;
            &lt;TargetState&gt;
                &lt;Condition Name="" Value="" /&gt;
                &lt;Condition Name="" Value="" /&gt;
            &lt;/TargetState&gt;
        &lt;/Target&gt;
    &lt;/Targets&gt;

    &lt;Static&gt;
        &lt;Settings Path="Multivariant"&gt;
            &lt;Setting Name="Enable" Value="1" /&gt;
        &lt;/Settings&gt;
        &lt;Settings Path="AutoDataConfig"&gt;
            &lt;Setting Name="Enable" Value="0" /&gt;
        &lt;/Settings&gt;
    &lt;/Static&gt;

    &lt;!-- Specify the Variant --&gt;
    &lt;Variant Name=""&gt;
        &lt;TargetRefs&gt;
            &lt;TargetRef Id="" /&gt;
        &lt;/TargetRefs&gt;

        &lt;Settings Path="CellCore/PerIMSI/$(__IMSI)/CellUX"&gt;
            &lt;!-- Use to show the 'View Internet APN' button in the cellular+SIM settings screen.
            Set to 0 or 'No' (to hide, default) or set to 1 or 'Yes' (to show). --&gt;
            &lt;Setting Name="ShowViewAPN" Value="" /&gt;
        &lt;/Settings&gt;
    &lt;/Variant&gt;

    --&gt;

    &lt;!-- Use for the per-device case

    &lt;Static&gt;
        &lt;Settings Path="CellCore/PerDevice/CellUX"&gt;
            &lt;!-- Use to show the 'View Internet APN' button in the cellular+SIM settings screen. Set to 0 or
            'No' (to hide, default) or set to 1 or 'Yes' (to show). --&gt;
            &lt;Setting Name="ShowViewAPN" Value="" /&gt;
        &lt;/Settings&gt;
    &lt;/Static&gt;

    --&gt;

&lt;/ImageCustomizations&gt;
</pre>

```

2. Specify an **Owner**.

3. Determine if you need to use the **per-IMSI** or **per-device** setting.

For the **per-IMSI** case:

- Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.

- b. Define settings for a **Variant**, which are applied if the associated target's conditions are met.
4. Set the value for `ShowViewAPN` to one of the following:

VALUE	DESCRIPTION
0 or 'No'	Hides the <b>View Internet APN</b> button in the <b>Cellular + SIM</b> settings screen. This is the default OS behavior.
1 or 'Yes'	Shows the <b>View Internet APN</b> button in the <b>Cellular + SIM</b> settings screen.

#### Testing:

1. Flash the build containing this customization to a device with a SIM.
2. Go to the **Cellular & SIM** screen in **Settings**.
3. Verify that the **View Internet APN** button is either hidden or visible depending on the value you set for `ShowViewAPN`.  
On a dual SIM device, verify if the **View Internet APN** button is hidden or visible depending on the `ShowViewAPN` setting value for each SIM.
4. If there is no data plan associated with the SIM or data is turned off, verify that the button is disabled.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Customizations for locale-based settings

10/2/2018 • 2 minutes to read • [Edit Online](#)

Describes the customizations that you can configure to optimize mobile devices for different regions they may ship to. Includes topics on shipping mobile devices to China.

## In this section

TOPIC	DESCRIPTION
<a href="#">Assistance for dialing international phone numbers</a>	Partners can turn off the international assist feature that helps users with the country codes needed for dialing international phone numbers.
<a href="#">China Type Approval requirement: app install prompts</a>	To meet China Type Approval (CTA) requirements for devices shipping in China, OEMs must show a notification dialog to alert users when the app being downloaded does certain things.
<a href="#">Contact management on the SIM (CN only)</a>	Partners can specify that users should be able to read, edit, delete, import, and export contact information on their SIM (basic SIM, USIM, or RUIM). This customization is only available for devices sold in China.
<a href="#">Disable NITZ or daylight saving time</a>	OEMs can configure Network Identity and Time Zone (NITZ) to handle daylight saving time appropriately for their market, or disable automatic setting of date and time completely.
<a href="#">Display location icon</a>	Partners can hide the location icon in the system tray if they choose.
<a href="#">Ignore NITZ information from LTE networks</a>	For mobile networks that can receive Network Identity and Time Zone (NITZ) information from multiple sources, partners can set the device to ignore the time received from an LTE network.
<a href="#">Microsoft Store for China</a>	For a Windows 10 Mobile device shipping in China, OEMs must specify that the device is intended for that market by setting the PhoneROMLanguage setting in DeviceTargetingInfo to the appropriate locale ID.
<a href="#">Mobile device languages</a>	Partners must select the set of available languages to include on the mobile device. Partners must also specify one of the included languages as the default device language.
<a href="#">Network Time Protocol support</a>	Use to automatically set the time using an NTP client in a mobile device that doesn't support NITZ, or when cellular data is not available.
<a href="#">Regional format</a>	Partners can specify the default country/region, regional format, pre-enabled keyboard, and speech languages for the device.

Topic	Description
<a href="#">Speech languages</a>	OEMs can specify the speech languages to include on the mobile device.
<a href="#">Default list of countries/regions</a>	OEMs can select the countries/regions to exclude from the default list shown in the mobile device's Country/Region list in the Settings screen.
<a href="#">Preferred system types for phone connectivity (CN only)</a>	OEMs can provide more control over the system types that their devices use to connect. This customization is only for China. OEMs should not set this customization unless required by the mobile operator.
<a href="#">Threshold for automatic time update</a>	For mobile networks that support Network Identity and Time Zone (NITZ), OEMs can specify the difference (in number of seconds) between the NITZ information and the current device time before a device time update is triggered.
<a href="#">Time zone priority list</a>	Beginning with Windows 10 Mobile, this customization is no longer necessary as the OS supports a location-based timezone detection service. However, to maintain backwards compatibility for some RPAL APIs that were previously released, an updated table of the timezone IDs is provided for your reference.
<a href="#">WAP browser support (CN and IN only)</a>	For phones that will ship in China and India, OEMs can add one preloaded WAP browser to the phone, which will automatically be launched when the user tries to open a WAP link. The WAP browser must be written as an application, and must go through the standard Microsoft Store submission process.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Assistance for dialing international phone numbers

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can turn off the international assist feature that helps users with the country codes needed for dialing international phone numbers. This customization is recommended when the device will be sold in a country or region that has multiple IDD (country exit code) values.

If the country or region has multiple IDD values but a default is set, international assist will occasionally be successful. The following lists show the countries and regions that are affected for placing or receiving calls.

- Multiple IDD values, No Default (rarely successful):

Belize, Brazil, Cambodia, Columbia, Indonesia, Israel, Korea, Maldives, Mauritius, Mongolia, New Caledonia, Singapore, Solomon Islands, Taiwan, Thailand, Uganda, Uruguay

- Multiple IDD values, default set (occasionally successful):

Australia, Costa Rica, Finland, Guatemala

If the international assist feature is turned off, it is also possible to override the MNC and MCC used for SMS. For more information, see [International assisted dialing for SMS](#).

**Constraints:** FirstVariationOnly

**Instructions:**

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="AssistedDialSetting"
    Description="Use to set the default state of the international assist feature that
    helps users with country codes needed for dialing international phone numbers."
    Owner=""
    OwnerType="OEM">
<Static>
    <Settings Path="Phone/PhoneSettings">
        <Setting Name="AssistedDialSetting" Value="" />
    </Settings>
</Static>
</ImageCustomizations>
```

2. Specify an **Owner**.

3. Set **AssistedDialSetting** to one of the following values:

VALUE	DESCRIPTION
0 or False	Turns off the international assist feature by default
1 or True	Turns on the international assist feature by default

By default, the international assist feature is turned off.

**Testing steps:**

1. Flash a build containing this customization to a device.
2. Go to the **Phone** settings screen.
3. Verify whether the **International assist** option is visible, and if so, whether it is turned on or off.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# CTA app install prompts

10/2/2018 • 2 minutes to read • [Edit Online](#)

To meet China Type Approval (CTA) requirements for devices shipping in China, OEMs must show a notification dialog to alert users when the app being downloaded does certain things.

**Note** This is a legacy mobile setting and is only a requirement for China. It works on phones being upgraded to Windows 10 Mobile, but we recommend that you use the [CountryAndRegion](#) Windows provisioning setting instead.

The dialog must be shown when the app being downloaded does any of the following:

- Invokes user data from a phone book
- Uses recording from the Microsoft Store
- Uses location data

**Constraints:** None

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>

<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="CTAAppInstallPrompts"
    Description="Use to show a notification dialog to alert users when the app being
downloaded invokes data from a
                                phone book, uses recording from the Windows Phone Store, or uses
location data.
                                This customization is only a requirement for China."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Zune/Settings">
            <Setting Name="RequireExtendedCapabilityPrompts" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set the value of `RequireExtendedCapabilityPrompts` to one of the following:

VALUE	DESCRIPTION
1 or 'Yes'	Shows a notification dialog when the user downloads an app from the Microsoft Store that supports the functionality described for the customization.

VALUE	DESCRIPTION
0 or 'No'	Disables the feature.

If the setting is not set, the feature is not enabled.

### Testing:

To fully test this customization, the phone must also be configured to use [Microsoft Store for China](#).

1. Flash the build containing this customization to a phone.
2. Go to the Microsoft Store and download and install an app that can invoke the phone book.
3. Open the app to access the device's native phone book.
4. Verify that a notification dialog shows up.
5. Download another app that uses recording or location data and verify that you see the notification appear when you first run the app.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Contact management on the SIM

10/2/2018 • 2 minutes to read • [Edit Online](#)

## IMPORTANT

This customization is only available for devices sold in China.

Partners can specify that users should be able to read, edit, delete, import, and export contact information on their SIM (basic SIM, USIM, or RUIM).

**Constraints:** ImageTimeOnly

## Instructions

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="SIMContactManagement"
    Description="Use to specify that users should be able to read, edit, delete,
import, and export contact information
on their SIM (basic SIM, USIM, or RUIM)."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="People/SIMContactManagement">
            <Setting Name="EnableSIMAddressBookAndExport" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set the value of `EnableSIMAddressBookAndExport` to one of the following values:

VALUE	DESCRIPTION
0 or 'False'	Disables the <b>export contacts to SIM/import SIM contacts</b> button in the <b>People</b> settings screen.
1 or 'True'	Enables the <b>export contacts to SIM/import SIM contacts</b> button in the <b>People</b> settings screen.

## Testing

1. Flash the build containing this customization to a device with a SIM.
2. Go to the **People** Hub and tap the + button to add a new contact.

3. Verify that you can see the **Create contact in** window, which includes an option for creating the contact in the **SIM**.
4. Choose where you would like to create the contact, and then add a new contact by filling in a name and phone number.
5. Go to the People **Settings** screen.
6. Verify that you can see both an **export contacts to SIM** button and an **import SIM contacts** button.
  - The **import SIM contacts** button will not be activated unless the SIM you used for testing already contains SIM contacts or the contact you created in step 4 was created in the SIM.
  - The **export contacts to SIM** button will not be activated unless there are contacts in your Microsoft account.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Disable NITZ or daylight saving time

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can configure NITZ to handle daylight saving time appropriately for their market, or disable automatic setting of date and time completely.

By default, the OS automatically sets the time and date by using Network Identity and Time Zone (NITZ). OEMs can configure NITZ to handle daylight saving time appropriately for their market, or disable automatic setting of date and time on the device completely if NITZ is not supported by the network.

In addition, if the device will be sold in a country or region that does not use daylight saving time, partners can disable it on the device. This will help the automatic time algorithm choose the correct time zone with greater accuracy.

**Constraints:** None

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="DisableNITZorDST"
    Description="Use to configure NITZ to handle DST or disable automatic date and time
setting if NITZ is not supported."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="AutomaticTime">

            <!-- Use to disable automatic date and time configuration -->
            <Setting Name="EnableAutomaticTime" Value="" />

            <!-- Use to disable automatic DST adjustment -->
            <Setting Name="DisableDaylightSavingsTime" Value="" />

        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. The following values are valid for `EnableAutomaticTime`:

VALUE	DESCRIPTION
0	Use to disable automatic time setting with NITZ.
1	Use to enable automatic time setting with NITZ.

4. The following values are valid for `DisableDaylightSavingsTime`:

VALUE	DESCRIPTION
0	Use to enable daylight saving time.
1	Use to disable daylight saving time.

5. Delete either of the `EnableAutomaticTime` or `DisableDaylightSavingsTime` settings if the feature does not need to be disabled.

#### Testing steps:

1. Flash a build containing this customization to a device with a SIM or UICC.
2. If you have disabled automatic time setting with NITZ, ensure that the date and time is not set automatically for the user during setup.  
If you have disabled NITZ, the default time zone shown during setup will come from the country/region default time zone list based on the current country/region setting. Also, the **date + time** screen in **Settings** will have fields for **Time zone**, **Date**, and **Time**, but no **Set automatically** toggle.
3. If daylight saving time has been disabled and automatic time is enabled, the device determines the most likely time zone by the current offset only.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Display location icon

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can hide the location icon in the system tray if they choose.

By default, the location icon in the system tray is displayed whenever an app requests the user's location. Users may override the setting in the **Location** settings screen.

**Constraints:** FirstVariationOnly

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="LocationIcon"
    Description="Use to configure the display of the location icon in the system tray."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="Shell/SystemTray/Location">
            <Setting Name="LocationIcon" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set the value of `LocationIcon` to one of the following:

VALUE	DESCRIPTION
1 or 'Enabled'	Displays the location icon in the system tray. This is the default.
0 or 'Disabled'	Hides the location icon in the system tray.

## Testing steps:

1. Flash a build containing this customization to a device.
2. Depending on the value you set for `LocationIcon`, verify if the location icon is displayed or is hidden in the system tray if an app that requests the user's location is launched.
3. Go to the **Settings > Location** screen. Change the value of the **Show icon** option and verify if the location icon is displayed or is hidden from the system tray based on the setting you chose.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Ignore NITZ information from LTE networks

10/2/2018 • 2 minutes to read • [Edit Online](#)

For mobile networks that can receive Network Identity and Time Zone (NITZ) information from multiple sources, partners can set the device to ignore the time received from an LTE network. Time received from a CDMA network is not affected.

**Constraints:** None

This customization supports: **per-IMSI** value, **per-device** value

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>

<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="NITZFiltering"
    Description="Use to set the phone to ignore the time received from an LTE network."
    Owner=""
    OwnerType="OEM">

    <!-- Use for the per-IMSI case

        <!-- Define the Targets -->
        <Targets>
            <Target Id="">
                <TargetState>
                    <Condition Name="" Value="" />
                    <Condition Name="" Value="" />
                </TargetState>
            </Target>
        </Targets>

        <Static>
            <Settings Path="Multivariant">
                <Setting Name="Enable" Value="1" />
            </Settings>
            <Settings Path="AutoDataConfig">
                <Setting Name="Enable" Value="0" />
            </Settings>
        </Static>

        <!-- Specify the Variant -->
        <Variant Name="">
            <TargetRefs>
                <TargetRef Id="" />
            </TargetRefs>

            <Settings Path="CellCore/PerIMSI/$(__IMSI)/General">
                <Setting Name="NitzFiltering" Value="0x10" />
            </Settings>
        </Variant>
    <!-->

    <!-- Use for the per-device case

        <Static>
            <Settings Path="CellCore/PerDevice/General">
                <Setting Name="NitzFiltering" Value="0x10" />
            </Settings>
        </Static>
    <!-->

</ImageCustomizations>

```

2. Specify an `Owner`.

3. Determine if you need to use the **per-IMSI** or **per-device** setting.

For the **per-IMSI** case:

- Define **Targets** or conditions for when a variant can be applied, such as keying off a SIM's MCC, MNC, and SPN.
- Define settings for a **Variant**, which are applied if the associated target's conditions are met.

4. Set the `Value` to 0x10.

The value specifies RIL\_SYSTEMTYPE\_LTE.

**Testing:**

Work with your mobile operator partner to test this customization on their network.

1. Flash the build containing this customization to a device capable of receiving NITZ information from both a CDMA network and LTE network.
2. Verify that the device only uses NITZ information received from the CDMA network.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Microsoft Store for China

10/2/2018 • 2 minutes to read • [Edit Online](#)

For a Windows 10 Mobile device shipping in China, OEMs must specify that the device is intended for that market by setting the **PhoneROMLanguage** setting in **DeviceTargetingInfo** to the appropriate locale ID. For example, for Chinese (China) the locale ID must be set to 0804. When enabled, users are routed to the Microsoft Store for China.

## Note

This customization is only a requirement for China.

**Constraints:** ImageTimeOnly

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="PhoneMetadataDeviceTargetingInfo"
    Description="Use to set phone metadata including the phone model name, OEM and
mobile operator name, hardware and software versions, and so on."
    Owner=""
    OwnerType="OEM">

    <!-- Define the Targets for the Variant -->
    <Targets>
        <Target Id="">
            <TargetState>
                <Condition Name="" Value="" />
                <Condition Name="" Value="" />
            </TargetState>
        </Target>
    </Targets>

    <Static>
        <Settings Path="Multivariant">
            <Setting Name="Enable" Value="1" />
        </Settings>
        <Settings Path="AutoDataConfig">
            <Setting Name="Enable" Value="0" />
        </Settings>
    </Static>

    <Static>
        <!-- These settings are ImageTimeOnly and will be put directly into the registry hive -->
        <Settings Path="DeviceInfo/Static">
            <Setting Name="PhoneManufacturer" Value="" />
            <Setting Name="PhoneROMVersion" Value="" />
            <Setting Name="PhoneHardwareRevision" Value="" />
            <Setting Name="PhoneSOCVersion" Value="" />
            <Setting Name="PhoneFirmwareRevision" Value="" />
            <Setting Name="PhoneRadioHardwareRevision" Value="" />
            <Setting Name="PhoneRadioSoftwareRevision" Value="" />
            <Setting Name="PhoneBootLoaderVersion" Value="" />
            <Setting Name="PhoneROMLanguage" Value="0804" />
            <Setting Name="PhoneHardwareVariant" Value="" />
        </Settings>
    </Static>

    <!-- Specify the Variant -->
    <Variant Name="">
        <TargetRefs>
            <TargetRef Id="" />
        </TargetRefs>

        <!-- These settings are FirstVariationOnly and can be configured at runtime potentially based on
SIM value -->
        <Settings Path="DeviceInfo/Variant">
            <Setting Name="PhoneMobileOperatorName" Value="" />
            <Setting Name="PhoneManufacturerModelName" Value="" />
            <Setting Name="PhoneMobileOperatorDisplayName" Value="" />
            <Setting Name="PhoneSupportPhoneNumber" Value="" />
            <Setting Name="PhoneSupportLink" Value="" />
            <Setting Name="PhoneOEMSupportLink" Value="" />
            <Setting Name="PhonemodelName" Value="" />
        </Settings>
    </Variant>

</ImageCustomizations>

```

2. Specify an **Owner**.

3. Set `PhoneROMLanguage` to 0804 for China (Chinese).

If partners do not set the `PhoneROMLanguage` setting to a China locale ID, partners may not ship the device in China. For more information about all locale IDs (LCIDs) including Chinese LCIDs, see [Locale IDs assigned by Microsoft](#) on MSDN.

**Testing:**

1. Flash the build containing this customization to a device with a UICC.
2. From the device, launch the Microsoft Store and verify that you are routed to the store for China.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Mobile device languages

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners must select the set of available languages to include on the mobile device. Partners must also specify one of the included languages as the default device language.

Windows 10 Mobile provides support for many languages. See the detailed list in this topic for more information.

## Limitations and restrictions:

- Partners must include at least one device language.
- Microsoft recommends that OEMs include all of the supported device languages, but the OEM must abide by the space limitations for the partition layout.
- Partners cannot modify existing languages or add support for their own.

For more information about language customizations, see the overview [Set languages and locales](#).

**Constraints:** ImageTimeOnly

## Instructions

To modify the list of device languages, you must edit the **UserInterface** section in the OEMInput.xml file before building the device image.

The following device languages are supported.

DEVICE LANGUAGE	VALUE TO USE IN THE OEMINPUT.XML FILE
Afrikaans	af-ZA
Albanian	sq-AL
Amharic	am-ET
Arabic	ar-SA
Azerbaijani (Latin)	az-Latn-AZ
Bangla	bn-BD
Basque (Basque)	eu-ES
Belarusian	be-BY
Bulgarian	bg-BG

DEVICE LANGUAGE	VALUE TO USE IN THE OEMINPUT.XML FILE
Catalan	ca-ES
Chinese (Simplified)	zh-CN
Chinese (Traditional)	zh-TW
Croatian	hr-HR
Czech	cs-CZ
Danish	da-DK
Dutch	nl-NL
English (United Kingdom)	en-GB
English (United States)	en-US
Estonian	et-EE
Filipino	fil-PH
Finnish	fi-FI
French (Canada)	fr-CA
French (France)	fr-FR
Galician	gl-ES
German	de-DE
Greek	el-GR
Hausa (Latin)	ha-Latn-NG
Hebrew	he-IL

DEVICE LANGUAGE	VALUE TO USE IN THE OEMINPUT.XML FILE
Hindi	hi-IN
Hungarian	hu-HU
Icelandic	is-IS
Indonesian	id-ID
Italian	it-IT
Japanese	ja-JP
Kannada	kn-IN
Kazakh	kk-KZ
Khmer	km-KH
Korean	ko-KR
Lao	lo-LA
Latvian	lv-LV
Lithuanian	lt-LT
Macedonian	mk-MK
Malay	ms-MY
Malayalam	ml-IN
Norwegian (Bokmål)	nb-NO
Persian	fa-IR
Polish	pl-PL

DEVICE LANGUAGE	VALUE TO USE IN THE OEMINPUT.XML FILE
Portuguese (Brazil)	pt-BR
Portuguese (Portugal)	pt-PT
Romanian	ro-RO
Russian	ru-RU
Serbian (Latin)	sr-Latn-RS
Slovak	sk-SK
Slovenian	sl-SI
Spanish (Mexico)	es-MX
Spanish (Spain)	es-ES
Swahili	sw-KE
Swedish	sv-SE
Tamil	ta-IN
Telugu	te-IN
Thai	th-TH
Turkish	tr-TR
Ukrainian	uk-UA
Uzbek (Latin)	uz-Latn-UZ
Vietnamese	vi-VN

- **List of included device languages:** OEMs must include at least one device language. To include multiple device languages, add additional `Language` entries to the `UserInterface` section of the OEMInput.xml file. In the following example, English (US), Japanese, and Dutch are included as device languages.

```

<SupportedLanguages>
  <UserInterface>
    <Language>en-US</Language>
    <Language>ja-JP</Language>
    <Language>n1-NL</Language>
  </UserInterface>
  <Keyboard>
    <Language>en-US</Language>
  </Keyboard>
  <Speech>
    <Language>en-US</Language>
  </Speech>
</SupportedLanguages>

```

- **Default device language:** to define the default device language that the device will use when it is first turned on by the user, OEMs must define both a default device language and a default regional format. Both values must be specified in the OEMInput.xml file before building the device image.
  - To specify the default device language, edit the **BootUILanguage** entry in OEMInput.xml. This value must match one of your **Language** entries from the **UserInterface** section.
  - To specify the default regional format, edit the **BootLocale** entry in OEMInput.xml.
  - Expanding on the previous example, the following shows how to set Japanese as the default device language (**BootUILanguage**) and set the default regional format (**BootLocale**) to Japan.

```

<SupportedLanguages>
  <UserInterface>
    <Language>en-US</Language>
    <Language>ja-JP</Language>
    <Language>n1-NL</Language>
  </UserInterface>
  <Keyboard>
    <Language>en-US</Language>
  </Keyboard>
  <Speech>
    <Language>en-US</Language>
  </Speech>
</SupportedLanguages>
<BootUILanguage>ja-JP</BootUILanguage>
<BootLocale>ja-JP</BootLocale>

```

#### NOTE

These two paired values either have to be the same or they must be associated. For more information, see the recommended values shown in the table in [Set languages and locales](#).

## Testing

1. Flash the build containing this customization to a device.
2. During initial device setup, do not change the device language.
3. Go to the **Language** screen in **Settings > Time & language**.
4. Look at the language list and verify that the default language is correct.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Network Time Protocol support

10/2/2018 • 3 minutes to read • [Edit Online](#)

Use to automatically set the time using an NTP client in a mobile device that doesn't support NITZ, or when cellular data is not available.

Mobile devices primarily rely on Network Identify and Time zone (NITZ), which is provided by the mobile operator, to automatically update the time on the device. When NITZ is available from the cellular network, there are no issues maintaining accurate time in devices. However, for devices that do not have a SIM or have had the SIM removed for some time, or for devices that have a SIM but NITZ is not supported, the device may run into issues maintaining accurate time on the device.

The OS includes support for Network Time Protocol (NTP), which enables devices to receive time when NITZ is not supported or when cellular data is not available. NTP gets the time by querying a server at a specified time interval. NTP is based on Coordinated Universal Time (UTC) and doesn't support time zone or daylight saving time so users will need to manually update the time zone after an update from NTP if users move between time zones.

For mobile devices that do not support NITZ and have NTP enabled, the user is required to select the time zone, date, and time during initial device setup before the Wi-Fi connections page. The Wi-Fi connection requires certificate validation, which needs accurate time.

If NTP is enabled, the first NTP query happens post-shell ready. After that, the default regular sync interval then applies.

**Constraints:** None

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="NTPSettings"
    Description="Use to automatically set the time, using an NTP client, in a Windows
    Phone device that
        doesn't support NITZ or when cellular data is not available."
    Owner=""
    OwnerType="OEM">

    <Static>
        <Settings Path="AutomaticTime">
            <Setting Name="NTPEnabled" Value="" />
            <Setting Name="NTPRegularSyncInterval" Value="" />
            <Setting Name="NTPRetryInterval" Value="" />
            <Setting Name="NTPServers" Value="" />
        </Settings>
    </Static>
</ImageCustomizations>
```

2. Specify an `Owner`.

3. To enable or disable the NTP client, set `NTPEnabled` to one of the following values:

VALUE	DESCRIPTION
0 or 'Disabled'	Disables the NTP client.
1 or 'Enabled'	Enables the NTP client.

**\*\*Note\*\***

Microsoft recommends explicitly setting a value for `NTPEnabled` depending on the user experience you want to enable or requirements you need to meet.

1. To set the regular sync interval, in hours, set `NTPRegularSyncInterval` to a value between 1 and 168 hours (inclusive). The default sync interval value is 12 hours.
2. To set the retry interval, in hours, if the regular sync fails, set `NTPRetryInterval` to a value between 1 and 24 hours (inclusive).
3. To enumerate the NTP source server(s) used by the NTP client, set the value for `NTPServer`. For example, the value can be `ntpserver1.contoso.com;ntpserver2.fabrikam.com;ntpserver3.contoso.com` and so on. The default NTP source server value `time.windows.com`.

### Testing:

1. Flash the build containing this customization to a device that does not support NITZ nor has a cellular data connection.
2. During initial device setup, verify that you see the **Time and region** screen. Set the correct time zone, date, and time for the device.
3. Verify that the Wi-Fi connection screen shows up after the **Time and region** screen. Connect to a Wi-Fi network so that the NTP client can connect to the NTP source server.
4. If you enabled NTP support, and depending on the values that you set for the regular sync interval, verify that the time on the device remains accurate after the sync interval has been reached. If the sync fails, verify if the correct time is set after the retry interval has passed.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Regional format

10/2/2018 • 3 minutes to read • [Edit Online](#)

Partners can specify the default country/region, regional format, pre-enabled keyboard, and speech languages for the device.

**Constraints:** ImageTimeOnly

**Instructions:**

The following table shows the country/region associated with the **BootLocale** values that you can use.

BOOTLOCALE VALUE	COUNTRY/REGION
af-ZA	South Africa
am-ET	Ethiopia
ar-AE	United Arab Emirates
ar-BH	Bahrain
ar-DZ	Algeria
ar-EG	Egypt
ar-IQ	Iraq
ar-JO	Jordan
ar-KW	Kuwait
ar-LB	Lebanon
ar-LY	Libya
ar-MA	Morocco
ar-OM	Oman
ar-QA	Qatar

BOOTLOCALE VALUE	COUNTRY/REGION
ar-SA	Saudi Arabia
ar-SY	Syria
ar-TN	Tunisia
ar-YE	Yemen
arn-CL	Chile
as-IN	India
az-Cyrl-AZ	Azerbaijan
az-Latn-AZ	Azerbaijan
ba-RU	Russia
be-BY	Belarus
bg-BG	Bulgaria
bn-NG	Nigeria
bn-BD	Bangladesh
bn-IN	India
bo-CN	China
br-FR	France
bs-Cyrl-BA	Bosnia and Herzegovina
bs-Latn-BA	Bosnia and Herzegovina
ca-ES	Spain

BOOTLOCALE VALUE	COUNTRY/REGION
ca-ES-valencia	Spain
chr-Cher-US	United States
co-FR	France
cs-CZ	Czech Republic
cy-GB	United Kingdom
da-DK	Denmark
de-AT	Austria
de-CH	Switzerland
de-DE	Germany
de-LI	Liechtenstein
de-LU	Luxembourg
dsb-DE	Germany
dv-MV	Maldives
dz-BT	Bhutan
el-GR	Greece
en-AU	Australia
en-BZ	Belize
en-CA	Canada
en-GB	United Kingdom

<b>BOOTLOCALE VALUE</b>	<b>COUNTRY/REGION</b>
en-HK	Hong Kong S.A.R.
en-ID	Indonesia
en-IE	Ireland
en-IN	India
en-JM	Jamaica
en-MY	Malaysia
en-NZ	New Zealand
en-PH	Philippines
en-SG	Singapore
en-TT	Trinidad and Tobago
en-US	United States
en-ZA	South Africa
en-ZW	Zimbabwe
es-AR	Argentina
es-BO	Bolivia
es-CL	Chile
es-CO	Colombia
es-CR	Costa Rica
es-CU	Cuba

BOOTLOCALE VALUE	COUNTRY/REGION
es-DO	Dominican Republic
es-EC	Ecuador
es-ES	Spain
es-GT	Guatemala
es-HN	Honduras
es-MX	Mexico
es-NI	Nicaragua
es-PA	Panama
es-PE	Peru
es-PR	Puerto Rico
es-PY	Paraguay
es-SV	El Salvador
es-US	United States
es-UY	Uruguay
es-VE	Bolivarian Republic of Venezuela
et-EE	Estonia
eu-ES	Spain
fa-IR	Iran
ff-Latn-SN	Senegal

BOOTLOCALE VALUE	COUNTRY/REGION
ff-NG	Nigeria
fi-FI	Finland
fil-PH	Philippines
fo-FO	Faroe Islands
fr-BE	Belgium
fr-CA	Canada
fr-CD	Congo (DRC)
fr-CH	Switzerland
fr-CI	Côte d'Ivoire
fr-CM	Cameroon
fr-FR	France
fr-HT	Haiti
fr-LU	Luxembourg
fr-MA	Morocco
fr-MC	Monaco
fr-ML	Mali
fr-RE	Reunion
fr-SN	Senegal
fy-NL	Netherlands

BOOTLOCALE VALUE	COUNTRY/REGION
ga-IE	Ireland
gd-GB	United Kingdom
gl-ES	Spain
gn-PY	Paraguay
gsw-FR	France
gu-IN	India
ha-Latn-NG	Nigeria
haw-US	United States
he-IL	Israel
hi-IN	India
hr-BA	Bosnia and Herzegovina
hr-HR	Croatia
hsb-DE	Germany
hu-HU	Hungary
hy-AM	Armenia
ibb-NG	Nigeria
id-ID	Indonesia
ig-NG	Nigeria
ii-CN	China

BOOTLOCALE VALUE	COUNTRY/REGION
is-IS	Iceland
it-CH	Switzerland
it-IT	Italy
iu-Cans-CA	Canada
iu-Latn-CA	Canada
ja-JP	Japan
kk-KZ	Kazakhstan
kl-GL	Greenland
km-KH	Cambodia
kn-IN	India
ko-KR	Korea
kok-IN	India
kr-NG	Nigeria
ks-Deva-IN	India
ku-Arab-IQ	Iraq
ky-KG	Kyrgyzstan
lb-LU	Luxembourg
lo-LA	Laos
lt-LT	Lithuania

BOOTLOCALE VALUE	COUNTRY/REGION
lv-LV	Latvia
mi-NZ	New Zealand
mk-MK	Macedonia, Former Yugoslav Republic of
ml-IN	India
mn-MN	Mongolia
mn-Mong-CN	China
mn-Mong-MN	Mongolia
mni-IN	India
moh-CA	Canada
mr-IN	India
ms-BN	Brunei
ms-MY	Malaysia
mt-MT	Malta
my-MM	Myanmar
nb-NO	Norway
ne-IN	India
ne-NP	Nepal
nl-BE	Belgium
nl-NL	Netherlands

BOOTLOCALE VALUE	COUNTRY/REGION
nn-NO	Norway
nso-ZA	South Africa
oc-FR	France
om-ET	Ethiopia
or-IN	India
pa-Arab-PK	Pakistan
pa-IN	India
pl-PL	Poland
prs-AF	Afghanistan
ps-AF	Afghanistan
pt-BR	Brazil
pt-PT	Portugal
qps-Latn-x-sh	Jamaica
qps-ploc	United States
qps-ploca	Japan
qps-plocm	Saudi Arabia
quc-Latn-GT	Guatemala
quz-BO	Bolivia
quz-EC	Ecuador

BOOTLOCALE VALUE	COUNTRY/REGION
quz-PE	Peru
rm-CH	Switzerland
ro-MD	Moldova
ro-RO	Romania
ru-MD	Moldova
ru-RU	Russia
rw-RW	Rwanda
sa-IN	India
sah-RU	Russia
sd-Arab-PK	Pakistan
sd-Deva-IN	India
se-FI	Finland
se-NO	Norway
se-SE	Sweden
si-LK	Sri Lanka
sk-SK	Slovakia
sl-SI	Slovenia
sma-NO	Norway
sma-SE	Sweden

BOOTLOCALE VALUE	COUNTRY/REGION
smj-NO	Norway
smj-SE	Sweden
smn-FI	Finland
sms-FI	Finland
so-SO	Somalia
sq-AL	Albania
sr-Cyrl-BA	Bosnia and Herzegovina
sr-Cyrl-CS	Serbia & Montenegro (Former) (Old Code)
sr-Cyrl-ME	Montenegro
sr-Cyrl-RS	Serbia
sr-Latn-BA	Bosnia and Herzegovina
sr-Latn-CS	Serbia & Montenegro (Former) (Old Code)
sr-Latn-ME	Montenegro
sr-Latn-RS	Serbia
st-ZA	South Africa
sv-FI	Finland
sv-SE	Sweden
sw-KE	Kenya
syr-SY	Syria

BOOTLOCALE VALUE	COUNTRY/REGION
ta-IN	India
ta-LK	Sri Lanka
te-IN	India
tg-Cyrl-TJ	Tajikistan
th-TH	Thailand
ti-ER	Eritrea
ti-ET	Ethiopia
tk-TM	Turkmenistan
tn-BW	Botswana
tn-ZA	South Africa
tr-TR	Turkey
ts-ZA	South Africa
tt-RU	Russia
tzm-Arab-MA	Morocco
tzm-Latn-DZ	Algeria
tzm-Tfng-MA	Morocco
ug-CN	China
uk-UA	Ukraine
ur-IN	India

BOOTLOCALE VALUE	COUNTRY/REGION
ur-PK	Pakistan
uz-Cyrl-UZ	Uzbekistan
uz-Latn-UZ	Uzbekistan
ve-ZA	South Africa
vi-VN	Vietnam
wo-SN	Senegal
xh-ZA	South Africa
yo-NG	Nigeria
zh-CN	China
zh-HK	Hong Kong S.A.R.
zh-MO	Macao S.A.R.
zh-SG	Singapore
zh-TW	Taiwan
zu-ZA	South Africa

**To set the default regional format:** The OEM must edit the **BootLocale** section of the OEMInput.xml file before building the device image.

The following example demonstrates how to set the default regional format to Japan (ja-JP) in the OEMInput.xml file.

```
<SupportedLanguages>
  <UserInterface>
    <Language>en-US</Language>
    <Language>ja-JP</Language>
    <Language>nl-NL</Language>
  </UserInterface>
  <Keyboard>
    <Language>en-US</Language>
  </Keyboard>
  <Speech>
    <Language>en-US</Language>
  </Speech>
</SupportedLanguages>
<BootUILanguage>ja-JP</BootUILanguage>
<BootLocale>ja-JP</BootLocale>
```

### Testing steps:

1. Flash the build containing this customization to a device.
2. During initial device setup, do not change the device language.
3. Go to the **Region** screen in **Settings > Time & language**.
4. Look at the **Country/Region** field to verify that the country or region is set according to the value you specified in the OEMInput.xml file.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Speech languages

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can specify the speech languages to include on the mobile device.

- The following 7 languages are supported for speech recognition and text-to-speech: en-US, en-GB, zh-CN, de-DE, fr-FR, it-IT, es-ES

These 7 languages are available on clean installs and can also be downloaded for devices running Windows 10 for desktop editions (Home, Pro, Enterprise, and Education) and Windows 10 Mobile.

- Additionally, Windows 10 includes 9 more languages for text-to-speech only: es-MX, en-IN, ko-KR, pt-BR, ja-JP, pl-PL, ru-RU, zh-TW, and zh-HK.

These 9 text-to-speech only languages are available on clean installs only and cannot be downloaded.

Users can install the 7 new speech languages from the **speech** screen in **Settings**, if they are not included on the device by default.

## Instructions:

For more information about language customizations, see the overview [Set languages and locales](#).

To modify the list of speech languages, you must edit the **Speech** section of the OEMInput.xml file before building the device image. You can omit the speech entirely, in which case the user will have to download languages before they can use the speech functionality on the device. To include speech, select one or more of the following options.

SPEECH LANGUAGE	VALUE TO USE IN THE OEMINPUT.XML FILE
English (United Kingdom)	en-GB
English (United States)	en-US
French	fr-FR
German	de-DE
Italian	it-IT
Spanish (Spain)	es-ES
Simplified Chinese	zh-CN
English (India) - TTS only	en-IN
Japanese - TTS only	ja-JP

SPEECH LANGUAGE	VALUE TO USE IN THE OEMINPUT.XML FILE
Korean - TTS only	ko-KR
Polish - TTS only	pl-PL
Portuguese (Brazil) - TTS only	pt-BR
Russian - TTS only	ru-RU
Spanish (Mexico) - TTS only	es-MX
Traditional Chinese (Hong Kong SAR) - TTS only	zh-HK
Traditional Chinese (Taiwan) - TTS only	zh-TW

To include one speech language, add one **Language** entry to the **Speech** section of the OEMInput.xml file as shown in the following example:

```
<SupportedLanguages>
  <UserInterface>
    <Language>en-US</Language>
  </UserInterface>
  <Keyboard>
    <Language>en-US</Language>
  </Keyboard>
  <Speech>
    <Language>en-US</Language>
  </Speech>
</SupportedLanguages>
```

To include multiple speech languages, add additional **Language** entries to the **Speech** section of the OEMInput.xml file as shown in the following sample.

```
<SupportedLanguages>
  <UserInterface>
    <Language>en-US</Language>
  </UserInterface>
  <Keyboard>
    <Language>en-US</Language>
  </Keyboard>
  <Speech>
    <Language>en-US</Language>
    <Language>pt-BR</Language>
    <Language>ru-RU</Language>
  </Speech>
</SupportedLanguages>
```

### Testing Steps:

1. Flash the build containing this customization to a device.
2. Go to the **speech** screen in **Settings**.

3. Verify that the list of speech languages installed on the device is correct.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Default list of countries/regions

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can select the countries/regions to exclude from the default list shown in the mobile device's **Country/Region** list in the **Settings** screen.

**Constraints:** None

**Instructions:**

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="DefaultRegionsList"
    Description="Use to specify list of countries/regions to exclude from
Country/Region list."
    Owner=""
    OwnerType="OEM">

    <Static>
        <Settings Path="Globalization">
            <Setting Name="ExcludedNations" Value="" />
        </Settings>
    </Static>
</ImageCustomizations>
```

2. Specify an `Owner`.
3. Set the value for `ExcludedNations` to specify the countries/regions that should not show up on the device's **Country/Region** list in the **Region** settings page.

You can specify the value as a list of semicolon-delimited ISO-3166-1 Alpha2 character codes (no spaces) that should be excluded when enumerating all supported countries with **EnumSystemGeoID**. The entire string must not be larger than 255 characters. The value must have a maximum of 85 codes only. Note that some ISO-3166-1 ALPHA2 codes cover multiple GeolIDs.

The value for `ExcludedNations` must follow this format:  
"IO;SJ;AQ;BV;CX;CC;HM;NF;MP;PN;GS;TK;WF;BL;UM"

**Testing steps:**

1. Flash the build containing this customization to a device.
2. After device setup, go to the **Region** screen in **Settings**.
3. .
4. Look at the **Country/Region** list to verify that the countries or regions you specified in `ExcludedNations` are not showing up on the list.

## Related topics

[Prepare for Windows mobile development](#)

## Customization answer file overview

# Preferred system types for phone connectivity

10/2/2018 • 3 minutes to read • [Edit Online](#)

## IMPORTANT

This customization is only for China. OEMs should not set this customization unless required by the mobile operator.

OEMs can provide more control over the system types that their devices use to connect by: mapping an ICCID or IIN to one radio (regardless of which SIM is chosen), specifying a list of MCC/MNCs that the MO wishes to limit, and/or restricting the second slot in a dual SIM device.

For mobile operators that require more control over the system types that their phones use to connect to the mobile operators' networks, OEMs can:

- Map a partial ICCID or Industry Identification Number (IIN) to the faster radio regardless of which SIM card is chosen for data connectivity.
- Specify the MCC and MNC of other specific operators that the main mobile operator wishes to limit. If the UICC's MCC and MNC matches any of the pairs that OEMs can specify for the operator, a specified RIL system type will be removed from the UICC regardless of its app types, slot position, or executor mapping.
- Restrict the second slot in a dual SIM device regardless of what apps or executor mapping the second slot is associated with.

**Constraints:** None

## Instructions

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="PreferredSystemTypesForPhoneConnectivity"
    Description="Use to provide a mobile operator more control over the system types
    that their phones
        use to connect to the operator's network."
    Owner=""
    OwnerType="OEM">

    <Static>
        <Settings Path="CellCore/PerDevice/General">

            <Setting Name="OperatorPreferredForFasterRadio" Value="" />

            <Setting Name="OperatorListForExcludedSystemTypes" Value="" />
            <Setting Name="ExcludedSystemTypesPerOperator" Value="" />

            <Setting Name="Slot2ExcludedSystemTypes" Value="" />

        </Settings>
    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. To map a partial ICCID or the IIN to the faster radio regardless of which SIM card is chosen for data connectivity, set the value of `OperatorPreferredForFasterRadio` to match the IIN or the ICCID, up to 7 digits, of the preferred operator.
4. To allow an operator to specify the MCC and MNC of other specific operators that they wish to limit, set the following settings:

- a. Set the value of the `OperatorListForExcludedSystemTypes` setting a comma separated list of MCC:MNC pairs for which the system types should be restricted.

For example, the value can be set to `310:026,310:030` to restrict operators with an MCC:MNC of 310:026 and 310:030.

- b. Set the value of the `ExcludedSystemTypesPerOperator` setting to match the system type to be excluded from the SIM cards that match the MCC:MNC pairs you listed in `OperatorListForExcludedSystemTypes`.

For example, a value of `0x8` specifies `RIL_SYSTEMTYPE_UMTS` (3G) while `0x10` specifies `RIL_SYSTEMTYPE_LTE` (4G). To exclude more than one system type, perform a bitwise **OR** operation on the radio technologies you want to exclude. For example, a bitwise **OR** operation on `RIL_SYSTEMTYPE_LTE` (4G) and `RIL_SYSTEMTYPE_UMTS` (3G) results in the value `11000` (binary) or `0x18` (hexadecimal). In this case, the `ExcludedSystemTypesPerOperator` value must be set to `0x18` to limit the matching MCC:MNC pairs to 2G.

5. To allow an operator to simply restrict the second slot in a dual SIM device regardless of what apps or executor mapping the second slot is associated with, set the value of `Slot2ExcludedSystemTypes` to the system types to be excluded from the SIM cards inserted in Slot 2.

For example, a value of `0x8` specifies `RIL_SYSTEMTYPE_UMTS` (3G) while `0x10` specifies `RIL_SYSTEMTYPE_LTE` (4G). To exclude more than one system type, perform a bitwise **OR** operation on the radio technologies you want to exclude. For example, a bitwise **OR** operation on `RIL_SYSTEMTYPE_LTE` (4G) and `RIL_SYSTEMTYPE_UMTS` (3G) results in the value `11000` (binary) or `0x18` (hexadecimal). In this case, any SIM inserted in Slot 2 will be limited to 2G.

## Testing

1. Work with your mobile operator to obtain the partial ICCID or the IIN, the list of MCC and MNC values that they wish to limit, and the system types that they wish to restrict.
2. Flash the build containing this customization to a dual SIM phone.
3. Depending on which settings you set to provide the mobile operator more control over the system types that their phones use to connect to the network, test each scenario to make sure that the device behaves as expected. With the settings in this customization, verify that you don't see the restricted mobile operators able to use any of the restricted RIL system types.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Threshold for automatic time update

10/2/2018 • 2 minutes to read • [Edit Online](#)

For mobile networks that support Network Identity and Time Zone (NITZ), OEMs can specify the difference (in number of seconds) between the NITZ information and the current device time before a device time update is triggered. When set, the OS updates the device time if the time difference is larger than the value specified by the OEM.

The threshold must be a value between 1 and 59 seconds, inclusive.

**Constraints:** None

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="NetworkTimeUpdateThreshold"
    Description="Use to specify the difference (in seconds) between the NITZ
information and the current
device time before a device time update is triggered."
    Owner=""
    OwnerType="OEM">

    <Static>
        <Settings Path="AutomaticTime">
            <Setting Name="NetworkTimeUpdateThreshold" Value="" />
        </Settings>
    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set the value for `NetworkTimeUpdateThreshold` between 1 and 59 seconds (inclusive). This is equivalent to 0x1 and 0x3B (inclusive) in hexadecimal.

## Testing:

Flash the build containing this customization to a phone connected to a network that supports NITZ.

## Note

OEMs may need to configure the value a few times to determine the best threshold value.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Time zone priority list

10/2/2018 • 3 minutes to read • [Edit Online](#)

Beginning with Windows 10 Mobile, this customization is no longer necessary as the OS supports a location-based timezone detection service. However, to maintain backwards compatibility for some RPAL APIs that were previously released (such as **GetTimeZoneInformationID**, **SetTimeZoneInformationByID**, and so on), an updated table of the timezone IDs is provided for your reference.

# #

ID	Time zone
0x0	UTC-12 International Date Line West
0x6E	UTC-11 Coordinated Universal Time-11
0xC8	UTC-10 Hawaii
0x12C	UTC-09 Alaska
0x190	UTC-08 Pacific Time (US & Canada)
0x19A	UTC-08 Baja California
0x1F4	UTC-07 Mountain Time (US & Canada)
0x1FE	UTC-07 Chihuahua, La Paz, Mazatlan
0x208	UTC-07 Arizona
0x258	UTC-06 Saskatchewan
0x262	UTC-06 Central America
0x26C	UTC-06 Central Time (US & Canada)
0x276	UTC-06 Guadalajara, Mexico City, Monterrey
0x2BC	UTC-05 Eastern Time (US & Canada)
0x2C6	UTC-05 Bogota, Lima, Quito, Rio Branco
0x2D0	UTC-05 Indiana (East)
0x2DA	UTC-05 Chetumal
0x348	UTC-4:30 Caracas
0x320	UTC-04 Atlantic Time (Canada)
0x32A	UTC-04 Cuiaba
0x33E	UTC-04 Georgetown, La Paz, Manaus, San Juan
0x352	UTC-04 Asuncion
0x384	UTC-03:30 Newfoundland
0x334	UTC-03 Santiago
0x38E	UTC-03 Brasilia
0x398	UTC-03 Greenland
0x3A2	UTC-03 Montevideo
0x3AC	UTC-03 Cayenne, Fortaleza
0x3B6	UTC-03 Buenos Aires
0x3C0	UTC-03 Salvador
0x3E8	UTC-02 Mid-Atlantic - Old
0x3F2	UTC-02 Coordinated Universal Time-02
0x44C	UTC-01 Azores
0x456	UTC-01 Cabo Verde Is.
0x4B0	UTC Greenwich Mean Time: Dublin, Edinburgh, Lisbon, London
0x4BA	UTC Monrovia, Reykjavik
0x4C4	UTC Casablanca
0x4CE	UTC Coordinated Universal Time
0x514	UTC+01 Belgrade, Bratislava, Budapest, Ljubljana, Prague
0x51E	UTC+01 Sarajevo, Skopje, Warsaw, Zagreb
0x528	UTC+01 Brussels, Copenhagen, Madrid, Paris
0x532	UTC+01 West Central Africa
0x53C	UTC+01 Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna
0x546	UTC+01 Windhoek
0x550	UTC+02 Tripoli
0x578	UTC+02 E. Europe
0x582	UTC+02 Cairo
0x58C	UTC+02 Helsinki, Kyiv, Riga, Sofia, Tallinn, Vilnius
0x596	UTC+02 Athens, Bucharest
0x5A0	UTC+02 Jerusalem
0x5AA	UTC+02 Amman
0x5B4	UTC+02 Beirut

ID	Time zone
0x5BE	UTC+02 Harare, Pretoria
0x5C8	UTC+02 Damascus
0x5D2	UTC+02 Istanbul
0x5DC	UTC+03 Kuwait, Riyadh
0x5E6	UTC+03 Baghdad
0x5F0	UTC+03 Nairobi
0x5FA	UTC+02 Kaliningrad (RTZ 1)
0x604	UTC+03 Moscow, St. Petersburg, Volgograd (RTZ 2)
0x618	UTC+03 Minsk
0x60E	UTC+03:30 Tehran
0x640	UTC+04 Abu Dhabi, Muscat
0x64A	UTC+04 Baku
0x654	UTC+04 Yerevan
0x668	UTC+04 Tbilisi
0x672	UTC+04 Port Louis
0x67C	UTC+04 Izhevsk, Samara (RTZ 3)
0x65E	UTC+04:30 Kabul
0x6A4	UTC+05 Ekaterinburg (RTZ 4)
0x6AE	UTC+05 Ashgabat, Tashkent
0x6D6	UTC+05 Islamabad, Karachi
0x6B8	UTC+05:30 Chennai, Kolkata, Mumbai, New Delhi
0x6C2	UTC+05:30 Sri Jayawardenepura
0x6CC	UTC+05:45 Kathmandu
0x708	UTC+06 Astana
0x712	UTC+06 Novosibirsk (RTZ 5)
0x726	UTC+06 Dhaka
0x71C	UTC+06:30 Yangon (Rangoon)
0x776	UTC+07 Bangkok, Hanoi, Jakarta
0x76C	UTC+07 Krasnoyarsk (RTZ 6)
0x7D0	UTC+08 Beijing, Chongqing, Hong Kong, Urumqi
0x7DA	UTC+08 Irkutsk (RTZ 7)
0x7E4	UTC+08 Kuala Lumpur, Singapore
0x7EE	UTC+08 Taipei
0x7F8	UTC+08 Perth
0x802	UTC+08 Ulaanbaatar
0x80C	UTC+08:30 Pyongyang
0x834	UTC+09 Seoul
0x83E	UTC+09 Osaka, Sapporo, Tokyo
0x848	UTC+09 Yakutsk (RTZ 8)
0x852	UTC+09:30 Darwin
0x85C	UTC+09:30 Adelaide
0x898	UTC+10 Canberra, Melbourne, Sydney
0x8A2	UTC+10 Brisbane
0x8AC	UTC+10 Hobart
0x8B6	UTC+10 Vladivostok, Magadan (RTZ 9)
0x8C0	UTC+10 Guam, Port Moresby
0x906	UTC+10 Magadan
0x8FC	UTC+11 Solomon Is., New Caledonia
0x91A	UTC+11 Chokurdakh (RTZ 10)
0x960	UTC+12 Fiji
0x96A	UTC+12 Auckland, Wellington
0x974	UTC+12 Petropavlovsk-Kamchatsky - Old
0x97E	UTC+12 Coordinated Universal Time+12
0x988	UTC+12 Anadyr, Petropavlovsk-Kamchatsky (RTZ 11)
0x9C4	UTC+13 Nuku'alofa
0x64	UTC+13 Samoa
0xA28	UTC+14 Kiribati Island

**Constraints:** None

#### Instructions:

**Note** The following instructions have been provided for backwards compatibility only. You may set the values for the `TimeZonePriority` settings, but the OS will ignore the values as a location-based timezone detection service is used instead.

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="TimeZonePriorityList"
    Description="Use To specify a preference list for the most applicable time zones
relative to the UTC offset."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="AutomaticTime">

            <Setting Name="TimeZonePriority1" Value="" />
            <Setting Name="TimeZonePriority2" Value="" />
            <Setting Name="TimeZonePriority3" Value="" />
            <Setting Name="TimeZonePriority4" Value="" />
            <Setting Name="TimeZonePriority5" Value="" />

            <!-- You can use up to 15 time zones, but do not add or set unless you need it.
            <Setting Name="TimeZonePriority6" Value="" />
            <Setting Name="TimeZonePriority7" Value="" />
            <Setting Name="TimeZonePriority8" Value="" />
            <Setting Name="TimeZonePriority9" Value="" />
            <Setting Name="TimeZonePriority10" Value="" />
            <Setting Name="TimeZonePriority11" Value="" />
            <Setting Name="TimeZonePriority12" Value="" />
            <Setting Name="TimeZonePriority13" Value="" />
            <Setting Name="TimeZonePriority14" Value="" />
            <Setting Name="TimeZonePriority15" Value="" />
            -->

        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.
3. Specify the `value` keys using the IDs of the time zones appropriate for the locations in which the device will be sold.
4. Add additional priorities as necessary. You can add up to 15 time zones or specify up to `TimeZonePriority15`. For example, if you are only specifying up to 3 time zones, only include up to the `TimeZonePriority3` setting, set their values, and do not add the rest.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# WAP browser support (CN and IN only)

10/2/2018 • 2 minutes to read • [Edit Online](#)

For phones that will ship in China and India, OEMs can add one preloaded WAP browser to the phone, which will automatically be launched when the user tries to open a WAP link. The WAP browser must be written as an application, and must go through the standard Microsoft Store submission process.

**Constraints:** None

## Instructions:

1. Create a WAP browser application and ensure that the following settings are configured properly:
  - a. Add the **ID\_CAP\_NETWORKING** capability to the application manifest of the WAP browser application.
  - b. Add the following XML to the **App** element of the application manifest.

```
<Extensions>
    <Protocol Name="wap" Category="phone.protocol" TaskID="_default" NavUriFragment="uri=%s"/>
    <FileTypeAssociation Name="TestFileAssoc1" Category="phone.fileTypeAssociation"
TaskID="_default"
        NavUriFragment="fileID=%s">
        <DisplayName>Test Assoc1</DisplayName>
        <Logo>res://StartMenu!AppIconMail.png</Logo>
        <SupportedFileTypes>
            <FileType ContentType="text/vnd.wap.wml">.wml</FileType>
        </SupportedFileTypes>
    </FileTypeAssociation>
</Extensions>
```

You can replace the values for following elements:

- **FileTypeAssociation Name**
  - **DisplayName**
  - **Logo**
- c. The WAP browser application manifest file should look like the following XML.

```

<?xml version="1.0" encoding="UTF-8"?>
<Deployment AppPlatformVersion="8.0"
xmlns="http://schemas.microsoft.com/windowsphone/2009/deployment">
    <App xmlns="" Publisher="TestWAPApp" Description="Sample description" Author="TestWAPApp
author"
        Genre="apps.normal" Version="1.0.0.0" RuntimeType="Silverlight" Title="TestWAPApp"
        ProductID="{bca7dae7-f8c3-4dc2-9a98-d6bf62b81a29}">
        <IconPath IsResource="false" IsRelative="true">ApplicationIcon.png</IconPath>
        <Capabilities> <Capability Name="ID_CAP_NETWORKING"/> </Capabilities>
        <Extensions>
            <Protocol Name="wap" Category="phone.protocol" TaskID="_default"
NavUriFragment="uri=%s"/>
            <FileTypeAssociation Name="TestFileAssoc1" Category="phone.fileTypeAssociation"
                TaskID="_default" NavUriFragment="fileID=%s">
                <DisplayName>Test Assoc1</DisplayName>
                <Logo>res://StartMenu!AppIconMail.png</Logo>
                <SupportedFileTypes>
                    <FileType ContentType=" text/vnd.wap.wml ">.wml</FileType>
                </SupportedFileTypes>
            </FileTypeAssociation>
        </Extensions>
        <Tasks>
            <DefaultTask Name="_default" NavigationPage="MainPage.xaml"/>
        </Tasks>
        <Tokens>
            <PrimaryToken TaskName="_default" TokenID="CustomUriTargetManagedApp1Token">
                <TemplateType5>
                    <BackgroundImageURI IsResource="false"
IsRelative="true">Background.png</BackgroundImageURI>
                    <Count>0</Count>
                    <Title>TestWAPApp</Title>
                </TemplateType5>
            </PrimaryToken>
        </Tokens>
    </App>
</Deployment>

```

For more information about file and URI associations, see the Windows 10 SDK documentation.

2. After you have created the WAP browser application and gone through the Microsoft Store submission process to obtain your application license, follow these steps to preload the WAP browser application to the phone.
  - a. Create a customization answer file using the contents shown in the following code sample.

```

<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="WAPBrowser"
    Description="Use to preload a WAP browser to the phone, which will be
automatically launched when the user tries to open a WAP link."
    Owner=""
    OwnerType="OEM">

    <Static>
        <Applications>
            <Application Source="C:\Path\OEM_WAPBrowser.xap"
                License="C:\Path\OEM_WAPBrowser_License.xml"
                ProvXML="C:\Path\MPAP_OEM_WAPBrowser.provxml" />
        </Applications>
    </Static>
</ImageCustomizations>

```

- b. Specify an  Owner .

- c. Set the `Source` value to the path and file name of your .xap or .appx file. For example, `C:\Path\OEM_WAPBrowser.xap`.
- d. Set the `License` value to the path and file name of license file for your WAP browser app. For example, `C:\Path\OEM_WAPBrowser_License.xml`.
- e. Set the `ProvXML` value to the path and file name for your WAP browser app. For example, `C:\Path\MPAP_OEM_WAPBrowser provxml`.

**Testing:**

1. Flash the build containing this customization to a phone with a SIM or with a data connection over Wi-Fi.
2. Open Microsoft Edge and enter a WAP link. The WAP browser should open automatically and display your web page.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Customizations for accessibility settings

10/2/2018 • 2 minutes to read • [Edit Online](#)

Describes the customizations that you can configure to enhance accessibility on the mobile device.

## In this section

TOPIC	DESCRIPTION
<a href="#">Telecoil and TTY support for accessibility</a>	Partners can choose whether to show TTY and/or Telecoil options in the device settings.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Telecoil and TTY support for accessibility

10/2/2018 • 3 minutes to read • [Edit Online](#)

Partners can choose whether to show TTY and/or Telecoil options in the device settings.

The OS provides support for telecoil and TTY devices. The settings and options that can be configured for telecoil and TTY appear in the **ease of access** screen in **Settings**. By default, both the telecoil and TTY options are hidden.

## TTY

A TTY is a separate device that enables people who are deaf, hard of hearing, or speech-impaired to communicate by sending and receiving text. TTY support is required at both ends of the conversation.

Partners can display a TTY/TDD option in the **ease of access** screen in **Settings**. Partners must decide whether to display two choices (Off or Full), or four choices (Off, Full, HCO, or VCO). If the TTY option is visible, it should be set to off by default.

## Telecoil

A Telecoil is a supported hearing aid or implant that enables people who are deaf or hard of hearing to listen to audio from the device by using magnetic induction.

Partners can display a Telecoil option in the **ease of access** screen in **Settings**. If the Telecoil toggle is visible, it should be set to off by default.

**Constraints:** Atomic, FirstVariationOnly

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="TelecoilAndTTY"
    Description="Use to display and configure the options for Telecoil and TTY/TDD
option in the
        'ease of access' screen under Settings."
    Owner=""
    OwnerType="OEM">

<Static>

    <!-- This settings group is atomic so all settings must be configured -->
    <Settings Path="EaseOfAccessCPL/TTY">
        <Setting Name="ShowInControlPanel" Value="" />
        <Setting Name="CompactMode" Value="" />
        <Setting Name="Mode" Value="" />
    </Settings>

    <!-- This settings group is atomic so all settings must be configured -->
    <Settings Path="EaseOfAccessCPL/Telecoil">
        <Setting Name="ShowInControlPanel" Value="" />
        <Setting Name="Enabled" Value="" />
    </Settings>

</Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Configure the TTY/TDD settings and default values. The following settings group under `EaseOfAccessCPL/TTY` is atomic so all settings must be configured.

a. To hide or show the TTY/TDD option in the **ease of access** screen in **Settings**, set the value of `ShowInControlPanel` to one of the following:

VALUE	DESCRIPTION
1 or 'Show'	Shows the TTY/TDD option in the <b>ease of access</b> screen in <b>Settings</b> .
0 or 'Hide'	Hides the TTY/TDD option in the <b>ease of access</b> screen in <b>Settings</b> .

b. To show two (Off or Full) or four (Off, Full, HCO, or VCO) menu items for TTY/TDD modes, set the value of `CompactMode` to one of the following:

VALUE	DESCRIPTION
1 or 'Enabled'	Shows four choices (Off, Full, HCO, or VCO) for the TTY selection UI.
0 or 'Disabled'	Shows two choices (Off or Full) for the TTY selection UI.

c. To specify the default mode for the TTY/TDD option, set the value of `Mode` to one of the following:

VALUE	DESCRIPTION
0 or 'Off'	Sets Off as the default user value. Microsoft recommends that partners use this as the default user value.
1 or 'Full'	Sets Full as the default user value.
2 or 'HCO'	Sets Hearing Carry Over (HCO) as the default user value.
3 or 'VCO'	Sets Voice Carry Over (VCO) as the default user value.

4. Configure the telecoil settings and default value. The following settings group under `EaseOfAccessCPL/Telecoil` is atomic so all settings must be configured.

a. To hide or show the Telecoil option in the **ease of access** screen in **Settings**, set the value of `ShowInControlPanel` to one of the following:

VALUE	DESCRIPTION
1 or 'Show'	Shows the Telecoil option in the <b>ease of access</b> screen in <b>Settings</b> .
0 or 'Hide'	Hides the Telecoil option in the <b>ease of access</b> screen in <b>Settings</b> .

- b. To set the default user value for the Telecoil option, set the value of `Enabled` to one of the following:

VALUE	DESCRIPTION
1 or 'Enabled'	Telecoil is on by default.
0 or 'Disabled'	Telecoil is off by default. Microsoft recommends that partners use this as the default user value.

To enable Telecoil, use the two registry entries exactly as shown in the TelecoilAndTTY.pkg.xml file.

To enable TTY/TTD, use the `tty_UI` registry entry exactly as shown in the TelecoilAndTTY.pkg.xml file. To display two choices (Off or Full), leave the `compactMode` registry value set to 0. To display four choices (Off, Full, HCO, or VCO), set the compact mode registry value to 1, instead.

#### Testing:

1. Flash the build containing this customization to a device.
2. Go to the **ease of access** screen in **Settings**.
3. Verify that the TTY and/or Telecoil options are visible and the default options are set accordingly. If TTY is visible, ensure that the correct number of options are shown (2 or 4).

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Customizations for phone update settings

10/2/2018 • 2 minutes to read • [Edit Online](#)

Describes the customizations that determine how the mobile device handles updates.

## In this section

TOPIC	DESCRIPTION
<a href="#">Auto scan for phone updates</a>	OEMs can show or hide the auto scan for updates setting on the device.
<a href="#">Block using SD card for updates</a>	For devices that support an SD card, OEMs can either allow or block the use of the SD card for device updates.
<a href="#">Enable SD card override</a>	OEMs can use EnableSDCardOverride to use the SD card for device updates.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Auto scan for phone updates

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs can show or hide the auto scan for updates setting on the device. When auto scan is visible, users can see a checkbox in the **Settings > device update** screen to inform them when updates are available for their devices. When hidden, the device will always scan for updates and the user option is not visible.

**Constraints:** None

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="DisplayCheckForUpdates"
    Description="Use to show or hide the auto scan setting in the Settings > Phone
    Update screen."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="DeviceUpdate">
            <!-- Use to determine whether to show or hide the auto scan settings for device updates. Set the
            value to 0 or 'Hidden',
            or set to 1 or 'Visible'. -->
            <Setting Name="DisplayCheckForUpdates" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set the value of `DisplayCheckForUpdates` to one of the following:

VALUE	DESCRIPTION
0 or 'Hidden'	The device will always scan for updates and the <b>Tell me when updates are available for my phone</b> checkbox is not displayed.
1 or 'Visible'	The <b>Tell me when updates are available for my phone</b> checkbox is displayed in the <b>Settings &gt; phone update</b> screen.

## Testing steps:

1. Flash a build containing this customization to a device.
2. Go to the **Settings > phone update** screen.
3. Depending on the value you set for `DisplayCheckForUpdates`, verify whether the **Tell me when updates are**

**available for my phone** checkbox is visible or hidden.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Block using SD card for updates

10/2/2018 • 2 minutes to read • [Edit Online](#)

For devices that support an SD card, OEMs can either allow or block the use of the SD card for device updates.

By default, this customization is not set and the OS can use the SD card for updates. If there is enough space on the eMMC to download an update, the OS will use the eMMC instead of the SD card.

**Constraints:** None

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="BlockUsingSDCard"
    Description="Use to determine whether to block the use of the SD card for device
updates."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="DeviceUpdate">
            <Setting Name="BlockUsingSDCard" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.

3. Set the value of `BlockUsingSDCard` to one of the following:

VALUE	DESCRIPTION
0 or 'No'	<p>Do not block the use of the SD card for device updates.</p> <div style="border: 1px solid black; padding: 5px;"><p><b>Note</b></p><p>Make sure that your UEFI supports powering up the SD card on the UpdateOS.</p></div>
1 or 'Yes'	Block the use of the SD card for device updates.

## Testing steps:

1. Flash a build containing this customization to a device.
2. If you set `BlockUsingSDCard` to allow the use of the SD card for updates and your device supports an SD card, if space on the eMMC is not enough for the update, the OS will choose the SD card to use for the

update.

When there is an update available and you have synced and downloaded the update, verify whether the update was installed from the SD card.

3. When the update has been successfully installed, use GetDULogs.exe to check if the update was done through the SD card.

Make sure that your UEFI supports the powering up of the SD card on the UpdateOS.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Enable SD card override

10/2/2018 • 2 minutes to read • [Edit Online](#)

By default, using the SD card for device updates is disabled. OEMs who want to use the SD card for device updates must set **EnableSDCardOverride** to opt-in and re-enable updates using the SD card. However, if OEMs set **BlockUsingSDCard** in [Block using SD card for updates](#), the value set for **BlockUsingSDCard** takes precedence.

**Constraints:** None

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="EnableSDCardOverride"
    Description="Use to configure whether the SD card can be used for updates on the
device."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="DeviceUpdate">
            <!-- Set the value to 0 or 'No' (do not use the SD card for updates), or set to 1 or 'Yes' (use
the SD card for updates). -->
            <Setting Name="EnableSDCardOverride" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner`.
3. Set the value of `EnableSDCardOverride` to one of the following:

VALUE	DESCRIPTION
0 or 'No'	Block the use of the SD card for phone updates. This is the default OS value.
1 or 'Yes'	Enable use of the SD card for phone updates.

## Testing steps:

1. Flash a build containing this customization to a phone.
2. Fill the Data partition and leave only 11 to 100 MB of available space.
3. To verify if the SD card is used for updates, look for the following message in the Installation ARD:

**WARNING: Do not remove your SD card while the update installs.**
4. If you set `EnableSDCardOverride` to allow the use of the SD card for updates and your phone supports an SD

card, and `BlockUsingSDCard` is not enabled, verify that you're able to use the SD card for device updates.

5. If you set `EnableSDCardOverride` to allow the use of the SD card for updates and your phone supports an SD card, but `BlockUsingSDCard` is enabled, verify that you're not able to use the SD card for device updates.

To verify the update scenario, try adding a new keyboard in the **Settings > Keyboard > add keyboards** screen and then select a new keyboard to add. This scenario uses the same path as device updates but is faster and does not require uploading builds to the update preview server.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Customizations for USB settings

10/2/2018 • 2 minutes to read • [Edit Online](#)

Describes the customizations that you can configure when a USB is connected to the mobile device.

## In this section

TOPIC	DESCRIPTION
<a href="#">Enable the incompatible charger notification</a>	Partners can set the device to display a warning when the user connects the device to an incompatible charging source.
<a href="#">Enable the data connection prompt</a>	Partners can set the device to display a dialog that asks for permission to enable the data connection when the user connects the device to a host computer via a USB cable.
<a href="#">Hide the weak charger notification option UI</a>	Partners can configure the device to hide the dialog that is displayed when the user connects the device to an incompatible charging source and hide the USB setting that notifies the user when the device is connected to a slower charger.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Enable the incompatible charger notification

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can set the device to display a warning when the user connects the device to an incompatible charging source. This warning is intended to notify users that their device may take longer to charge or may not charge at all with the current charging source.

An incompatible charging source is one that does not behave like one of the following port types as defined by the *USB Battery Charging Specification, Revision 1.2*, available on the [USB.org](#) website:

- Charging downstream port
- Standard downstream port
- Dedicated charging port

This setting is also available to users in the **USB** settings screen.

**Constraints:** ImageTimeOnly

**Instructions:**

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>

<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="USBIncompatibleCharger"
    Description="Use to display a dialog notifies the user when the phone is connected
to an incompatible charger."
    Owner=""
    OwnerType="OEM">

    <Static>
        <Settings Path="WeakCharger">
            <Setting Name="NotifyOnWeakCharger" Value="" />
        </Settings>
    </Static>

</ImageCustomizations>
```

2. Specify an `Owner` value in the customization answer file.

3. Set the `Value` to one of the following:

VALUE	DESCRIPTION
0 or 'Disable'	Do not display a dialog that notifies the user when the device is connected to an incompatible charger. This is the default value.
1 or 'Enable'	Display a dialog that notifies the user when the device is connected to an incompatible charger.

**Testing:**

1. Build an image that has this customization enabled, and flash this image to a device.
2. Connect the device to an incompatible charging source.
3. Confirm that the device displays a dialog that states that a weak or unsupported USB charger is connected.
4. Verify that the setting is also available to users in the **USB** settings screen.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Enable the data connection prompt

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can set the device to display a dialog that asks for permission to enable the data connection when the user connects the device to a host computer via a USB cable. Partners may need to enable this customization for certain markets.

By default, when the user connects the device to a host computer via a USB cable, the USB data connection is automatically enabled. This means that certain media files on the device, including pictures and music, are accessible through Windows Explorer on the host computer.

This setting is also available to users in the **USB** settings screen.

**Constraints:** None

## Instructions:

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="USBDataConnection"
    Description="Use to display a dialog that asks for permission to enable the data
connection when
    the user connects the device to a host computer via a USB cable."
    Owner=""
    OwnerType="OEM">

    <Static>
        <Settings Path="USBData">
            <Setting Name="PromptForDataConnection" Value="" />
        </Settings>
    </Static>

</ImageCustomizations>
```

2. Specify an `Owner` value in the customization answer file.

3. Set the `Value` to one of the following:

VALUE	DESCRIPTION
0	When the user connects the device to a host computer via a USB cable, do not display a dialog that asks for permission to enable the data connection. This is the default value.
1	When the user connects the device to a host computer via a USB cable, display a dialog that asks for permission to enable the data connection.

## Testing:

1. Build an image that has this customization enabled, and flash this image to a device.

2. Connect the device to a host computer using a USB cable.
3. Confirm that the device displays a dialog that asks for permission to enable the data connection.
4. Verify that the setting also appears in the **USB** settings screen.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Hide the weak charger notification option UI

10/2/2018 • 2 minutes to read • [Edit Online](#)

Partners can configure the device to hide the dialog that is displayed when the user connects the device to an incompatible charging source and hide the USB setting that notifies the user when the device is connected to a slower charger.

When this customization is configured, if the device is connected to an incompatible charger, the option to show the **Phone charging slowly** dialog is hidden, and the **Notify me if my mobile device is charging slowly over USB** toggle is hidden.

By default, the OS shows the weak charger notification option UI.

**Constraints:** ImageTimeOnly

**Instructions:**

1. Create a customization answer file using the contents shown in the following code sample.

```
<?xml version="1.0" encoding="utf-8" ?>
<ImageCustomizations xmlns="http://schemas.microsoft.com/embedded/2004/10/ImageUpdate"
    Name="USBHideWeakChargerNotificationUI"
    Description="Use to hide the weak charger notification option UI."
    Owner=""
    OwnerType="OEM">

    <Static>

        <Settings Path="WeakCharger">
            <Setting Name="HideWeakChargerNotifyOptionUI" Value="" />
        </Settings>

    </Static>

</ImageCustomizations>
```

2. Specify an `Owner` value in the customization answer file.
3. Set the value for `HideWeakChargerNotifyOptionUI` to one of the following values:

VALUE	DESCRIPTION
0 or 'False'	Shows the weak charger notification option UI. This is the default OS behavior.
1 or 'True'	Hides the weak charger notification option UI.

**Testing:**

1. Build an image that has this customization configured, and flash this image to a device.
2. Connect the device to an incompatible charging source.
3. If the weak charger notification UI is suppressed, verify the following behavior:

- Confirm that the device does not display the dialog that states that a weak or unsupported USB charger is connected.
- Verify that the **Notify me if my mobile device is charging slowly over USB** setting is also hidden to a user in the **USB** settings screen.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Registry values for mobile operator IDs

10/2/2018 • 28 minutes to read • [Edit Online](#)

Values to use for the mobile operator registry setting.

The following table contains the values to use for the mobile operator registry setting, **PhoneMobileOperatorName**, when the device is being provisioned for a single mobile operator. For more information about the registry setting, see [Phone metadata in DeviceTargetingInfo](#). If you are building a carrier-unlocked phone, use the topic [Registry values for carrier-unlocked phones](#) instead.

## Note

Although these mobile operator identifiers have been assigned, not all mobile operators in the table may be valid at this time. If you need to request for a new MOID, see [Requests for a new MOID](#).

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
AWC-AF	Afghan Wireless Communication Company	AWCC	Afghanistan
ETI-AF	Etisalat Afghanistan	Etisalat Afghanistan	Afghanistan
MTN-AF	MTN Afghanistan	MTN Afghanistan	Afghanistan
ROS-AF	Telecom Development Company Afghanistan Ltd.	ROSHAN	Afghanistan
AMC-AL	Albanian Mobile Communications	A M C MOBIL	Albania
EAG-AL	Eagle Mobile sh.a.	Eagle Mobile	Albania
PLU-AL	MOBILE 4 AL Sh.a	PLUS	Albania
VOD-AL	Vodafone Albania	vodafone	Albania
AMB-DZ	ATM MOBILIS	Mobilis	Algeria
DJE-DZ	Orascom Telecom Algerie Spa	Djezzy	Algeria
NED-DZ	Wataniya Telecom Algerie	Nedjma	Algeria

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>OOR-DZ</b>	Ooredoo Algeria	Ooredoo Algeria	Algeria
<b>BSK-AS</b>	American Samoa License, Inc.	Blue Sky Communications	American Samoa
<b>MAN-AD</b>	Servei De Tele. DAndorra	MOBILAND	Andorra
<b>MOV-AO</b>	Movicel Telecomunicações S.A.	Movicel Angola	Angola
<b>UNI-AO</b>	UNITEL S.a.r.l.	UNITEL	Angola
<b>CW0-AI</b>	Cable & Wireless (West Indies) Ltd. Anguilla	Cable & Wireless (West Indies) Anguilla	Anguilla
<b>APU-AG</b>	Antigua Public Utilities Authority-APUA	APUA PCS	Antigua and Barbuda
<b>CIN-AG</b>	Antigua Wireless Ventures Limited	Digicel Antigua & Barbuda	Antigua and Barbuda
<b>CW0-AG</b>	Cable & Wireless Caribbean Cellular (Antigua) Limited	Cable & Wireless	Antigua and Barbuda
<b>ART-AR</b>	Telecom Personal SA	Personal	Argentina
<b>CLA-AR</b>	AMX Argentina S.A.	CLARO ARGENTINA	Argentina
<b>FON-AR</b>	Telefonica Moviles Argentina S.A	Movistar	Argentina
<b>AMT-AM</b>	ArmenTel	ARMGSM	Armenia
<b>MTS-AM</b>	K Telecom CJSC	MTS Armenia	Armenia
<b>ORG-AM</b>	Orange Armenia CJSC	Orange	Armenia
<b>DIG-AW</b>	New Millenium Telecom Services (NMTS)	Digicel	Aruba

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>SET-AW</b>	SETAR (Servicio di Telecommunicacion di Aruba)	SETAR GSM	Aruba
<b>HUT-AU</b>	Hutchison 3G Australia Pty Limited	Hutchison 3G Australia Pty	Australia
<b>SIN-AU</b>	Singtel Optus Limited	YES OPTUS	Australia
<b>STR-AU</b>	Telstra Corporation Limited	Telstra MobileNet	Australia
<b>VOD-AU</b>	Vodafone Pacific Limited	vodafone	Australia
<b>AT1-AT</b>	Mobilkom Austria AG	A1	Austria
<b>BOB-AT</b>	bob	A1 MVNO	Austria
<b>HUT-AT</b>	Hutchison 3G Austria GmbH	3 AT	Austria
<b>ORG-AT</b>	Orange Austria Telecommunication GmbH	Orange	Austria
<b>TL2-AT</b>	Tele2 Austria	Tele2	Austria
<b>TMO-AT</b>	T-Mobile Austria GmbH	T-Mobile	Austria
<b>TRG-AT</b>	Telering Austria	Telering	Austria
<b>ACE-AZ</b>	Azercell Telecom LLC	AZERCELL GSM	Azerbaijan
<b>BKC-AZ</b>	Bakcell Ltd	BAKCELL GSM 2000	Azerbaijan
<b>NAR-AZ</b>	Azerfon LLC	Nar Mobile	Azerbaijan
<b>BAT-BS</b>	The Bahamas Telecommunications Company Ltd	The Bahamas Telecommunications Company	Bahamas, The

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>BTC-BH</b>	Bahrain Telecommunications Company	BATELCO	Bahrain
<b>VIV-BH</b>	STC Bahrain B.S.C	STC Bahrain	Bahrain
<b>ZAI-BH</b>	Zain Bahrain B.S.C	zain BH	Bahrain
<b>AKT-BD</b>	Axiata (Bangladesh) Limited.	Robi	Bangladesh
<b>BAN-BD</b>	Orascom Telecom Bangladesh Limited.	Banglalink	Bangladesh
<b>BMO-BD</b>	Teletalk Bangladesh Ltd	Teletalk	Bangladesh
<b>GRA-BD</b>	Grameenphone Ltd	Grameenphone	Bangladesh
<b>WAT-BD</b>	Warid Telecom International Ltd	Warid Telecom	Bangladesh
<b>CW0-BB</b>	Cable & Wireless Barbados Ltd.	Cable & Wireless (Barbados)	Barbados
<b>DIG-BB</b>	Digicel (Barbados) Limited	Digicel	Barbados
<b>LIF-BY</b>	Belarusian Telecommunications Network CJSC	life:)	Belarus
<b>MTS-BY</b>	JLLC Mobile TeleSystems	MTS	Belarus
<b>VEL-BY</b>	FE VELCOM	VELCOM	Belarus
<b>BAS-BE</b>	KPN Group Belgium NV/SA	KPN Group Belgium	Belgium
<b>MBS-BE</b>	Mobistar S.A. (Orange operates as Mobistar in Belgium)	Mobistar	Belgium

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>PRO-BE</b>	Belgacom SA/NV	PROXIMUS	Belgium
<b>TNT-BE</b>	Telenet Mobile	Mobistar	Belgium
<b>BTL-BZ</b>	Belize Telemedia Limited	Belize Telecommunications	Belize
<b>BBC-BJ</b>	Bell Benin Communications (BBCOM)	Bell Benin Communication BBCOM	Benin
<b>GLO-BJ</b>	Glomobile Benin Limited	GloBenin	Benin
<b>MTN-BJ</b>	Spacetel-Benin	MTN	Benin
<b>TLC-BJ</b>	Etisalat Benin S.A	TELECEL BENIN	Benin
<b>BDC-BM</b>	Bermuda Digital Communications Limited	CellularOne - Bermuda	Bermuda
<b>CIN-BM</b>	Telecommunications (Bermuda & West Indies) Ltd	Digicel Bermuda	Bermuda
<b>M3W-BM</b>	M3 Wireless Ltd	M3 Wireless	Bermuda
<b>M3W-BM</b>	M3 Wireless Ltd	M3 Wireless	Bermuda
<b>BMO-BT</b>	B-Mobile	B-Mobile	Bhutan
<b>TAS-BT</b>	Tashi InfoComm Limited	TashiCell	Bhutan
<b>EMO-BO</b>	Entel SA	Entel	Bolivia
<b>FON-BO</b>	Telefonica Celular De Bolivia S.A.	TELECEL BOLIVIA	Bolivia
<b>VIV-BO</b>	Nuevatel PCS De Bolivia SA	Nuevatel PCS De Bolivia	Bolivia

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>BHM-BA</b>	BH Telecom, Joint Stock Company, Sarajevo	BH Mobile	Bosnia and Herzegovina
<b>HTE-BA</b>	Public Enterprise Croatian Telecom Ltd.	ERONET	Bosnia and Herzegovina
<b>MTE-BA</b>	RS Telecommunications JSC Banja Luka	m:tel	Bosnia and Herzegovina
<b>MAS-BW</b>	Mascom Wireless (Pty) Limited	MASCOM	Botswana
<b>ORG-BW</b>	Orange (Botswana) Pty Limited	ORANGE	Botswana
<b>BRT-BR</b>	14 Brasil Telecom Celular S.A	Brasil Telecom Celular	Brazil
<b>CLA-BR</b>	CLARO S.A	Claro	Brazil
<b>CTB-BR</b>	CTBC Celular S.A.	CTBC Cellular	Brazil
<b>OIB-BR</b>	Oi Móvel	Telemar	Brazil
<b>SCT-BR</b>	Sercomtel Celular S/A	SERCOMTEL	Brazil
<b>TIM-BR</b>	TIM Celular S.A.	TIM BRASIL	Brazil
<b>VIV-BR</b>	Vivo S.A.	Vivo	Brazil
<b>BMO-BN</b>	B-Mobile Communications Sdn Bhd	b-mobile	Brunei
<b>DST-BN</b>	DataStream Technology	DTSCCom	Brunei
<b>EAD-BG</b>	Mobiltel EAD	M-Tel BG	Bulgaria
<b>GLO-BG</b>	Cosmo Bulgaria Mobile EAD	GLOBUL	Bulgaria

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>VIV-BG</b>	Bulgarian Telecommunications Company AD	vivacom	Bulgaria
<b>CBF-BF</b>	Celtel Burkina Faso	Zain Burkina Faso	Burkina Faso
<b>BDI-BI</b>	Telecel-Burundi Company	Telecel-Burundi Company	Burundi
<b>EWB-BI</b>	Econet Wireless Burundi PLC	Econet Wireless Burundi	Burundi
<b>ONA-BI</b>	ONATEL	ONATEL	Burundi
<b>SMA-BI</b>	LACELL SU	Smart Mobile	Burundi
<b>TEM-BI</b>	Africell PLC Company	AFRICELL PLC COMPANY	Burundi
<b>BEE-KH</b>	Sotelco Ltd.	Beeline-KH	Cambodia
<b>CAD-KH</b>	Cambodia Advance Communications Co. Ltd (CADCOMMS)	CADCOMMS	Cambodia
<b>MEF-KH</b>	VIETTEL (CAMBODIA) PTE., LTD	Metfone	Cambodia
<b>MTK-KH</b>	CamGSM	MOBITEL	Cambodia
<b>SMA-KH</b>	Latelz Co., Ltd	SMART MOBILE	Cambodia
<b>STA-KH</b>	APPLIFONE CO. LTD.	StarCell	Cambodia
<b>TMI-KH</b>	Hello Axiata Company Limited	hello	Cambodia
<b>MTN-CM</b>	MTN Cameroon Ltd	MTN Cameroon	Cameroon
<b>ORG-CM</b>	Orange Cameroun S.A.	Orange	Cameroon

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>BEL-CA</b>	Bell Mobility Inc.	Bell Mobility	Canada
<b>CHW-CA</b>	Chatrwireless	Chatr Wireless	Canada
<b>CTF-CA</b>	Cityfone	Cityfone	Canada
<b>FID-CA</b>	Microcell Telecommunications Inc (Fido)	FIDO	Canada
<b>KDO-CA</b>	Koodo	Koodo	Canada
<b>LUS-CA</b>	TELUS Communications Company	TELUS Communications	Canada
<b>MCW-CA</b>	Mobilicity Wireless	Mobilicity Wireless	Canada
<b>MTS-CA</b>	MTS Allstream Inc.	MTS	Canada
<b>PCM-CA</b>	PC Mobile	TELUS Communications	Canada
<b>PUB-CA</b>	Public Mobile	Public Mobile Canada	Canada
<b>ROG-CA</b>	Rogers Wireless Inc	Rogers Wireless	Canada
<b>SAT-CA</b>	SaskTel	SaskTel	Canada
<b>TST-CA</b>	TerreStar Solutions Inc	TerreStar Solutions	Canada
<b>VIR-CA</b>	Virgin Mobile Canada	Virgin Mobile Canada	Canada
<b>VTN-CA</b>	Videotron	Videotron	Canada
<b>WIN-CA</b>	Globalive Wireless LP	Globalive Wireless LP	Canada
<b>CMO-CV</b>	CVMoveL, S.A.	CVMOVEL	Cabo Verde

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>T00-CV</b>	T+ TELECOMUNICACÕES S.A.	T+ TELECOMUNICACOES	Cabo Verde
<b>LIM-KY</b>	Cable & Wireless (Cayman Islands) Limited	Cable & Wireless (Cayman Islands)	Cayman Islands
<b>NLR-CF</b>	Nationlink Telecom	Nationlink Telecom RCA	Central African Republic
<b>TLL-CF</b>	TELECEL CENTRAFRIQUE	TELECEL CENTRAFRIQUE	Central African Republic
<b>CEL-TD</b>	CelTel Tchad SA	Zain Chad	Chad
<b>CLA-CL</b>	CLARO CHILE SA	CLARO CHILE	Chile
<b>ENT-CL</b>	ENTEL TELEFONIA MOVIL	ENTEL TELEFONIA MOVIL	Chile
<b>FON-CL</b>	Telefonica Movil de Chile	TELEFONICA	Chile
<b>NXT-CL</b>	Nextel	Nextel	Chile
<b>VTR-CL</b>	VTR movil	VTR movil	Chile
<b>CMC-CN</b>	China Mobile	CHINA MOBILE	China
<b>CNT-CN</b>	China Telecom Corp. Ltd	China Telecom	China
<b>UNI-CN</b>	China Unicom	CHINA UNICOM GSM	China
<b>COM-CO</b>	Comunicacion Celular SA Comcel SA	Comunicacion Celular SA Comcel SA	Colombia
<b>ETB-CO</b>	Empresa de Telecomunicaciones de Bogotá S.A. ESP	Empresa de Telecomunicaciones de Bogotá S.A. ESP (ETB)	Colombia
<b>FON-CO</b>	Telefonica Moviles Colombia S.A.	movistar	Colombia

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>TIG-CO</b>	Colombia Movil SA	Tigo	Colombia
<b>HUR-KM</b>	Societe Nationale des Telecommunications (Comores Telecom)	HURI	Comoros
<b>AZU-CD</b>	EQUATEUR TELECOM CONGO S.A (ETC)	Azur-Congo	Congo (DRC)
<b>CCO-CD</b>	Congo Chine Telecoms	Congo Chine Telecoms	Congo (DRC)
<b>CEC-CD</b>	Celtel Congo	Zain Congo DRC	Congo (DRC)
<b>MTN-CD</b>	MTN CONGO S.A	Libertis Telecom	Congo (DRC)
<b>OAS-CD</b>	TIGO	tiGO	Congo (DRC)
<b>SCE-CD</b>	Supercell Sprl	Supercell	Congo (DRC)
<b>VOD-CD</b>	Vodacom Congo (RDC) sprl	VODACOM CONGO	Congo (DRC)
<b>WAR-CD</b>	Warid Congo SA	Warid Congo	Congo (DRC)
<b>KOK-CK</b>	Telecom Cook Islands	Telecom Cook Islands	Cook Islands
<b>FON-CR</b>	Telefonica Moviles Costa Rica	Moviestar	Costa Rica
<b>CLR-CR</b>	Claro	America Movil	Costa Rica
<b>ICE-CR</b>	I.C.E. (Instituto Costarricense de Electricidad)	I.C.E.	Costa Rica
<b>KOZ-CI</b>	Comium Ivory Coast Inc	KoZ	Côte d'Ivoire
<b>MOO-CI</b>	Atlantique Telecom Cote d'Ivoire	Moov	Côte d'Ivoire

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>MTN-CI</b>	MTN CÔTE D'IVOIRE S.A.	MTN	Côte d'Ivoire
<b>ORG-CI</b>	Orange CI	Orange CI	Côte d'Ivoire
<b>BON-HR</b>	bonbon	Telekom Croatia	Croatia
<b>TEL-HR</b>	Tele2 d.o.o za telekomunikacijske uluge	Tele2	Croatia
<b>TMO-HR</b>	Croatian Telecom Inc.	T-Mobile HR	Croatia
<b>TOM-HR</b>	Tomato	VIP-NET MVNO	Croatia
<b>VIP-HR</b>	VIPnet d.o.o.	VIP-NET	Croatia
<b>CYT-CY</b>	Cyprus Telecommunications Auth	Cytamobile-Vodafone	Cyprus
<b>MTN-CY</b>	MTN Cyprus Limited	MTN	Cyprus
<b>O2O-CZ</b>	Telefónica O2 Czech Republic a.s.	O2 - CZ	Czech Republic
<b>TMO-CZ</b>	T-Mobile Czech Republic a.s.	T-Mobile CZ	Czech Republic
<b>VOD-CZ</b>	Vodafone Czech Republic a.s.	OSKAR	Czech Republic
<b>DK3-DK</b>	Hi3G Denmark ApS	3 DK	Denmark
<b>DKT-DK</b>	TDC A/S	TDC Mobil	Denmark
<b>TLI-DK</b>	Telia	Telia	Denmark
<b>TOR-DK</b>	Telenor	Telenor	Denmark
<b>EVA-DJ</b>	Djibouti Telecom SA	Evatis	Djibouti

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>CIN-DM</b>	Wireless Ventures (Dominica) Ltd.	Digicel Dominica	Dominica
<b>CW0-DM</b>	Cable & Wireless Dominica Ltd.	Cable & Wireless Dominica	Dominica
<b>CLA-DO</b>	Compañia Dominicana de Telefonos, S.A.	CLARO GSM	Dominican Republic
<b>ORG-DO</b>	Orange Dominicana S.A.	ORANGE	Dominican Republic
<b>MOV-EC</b>	Telefonica Moviles Ecuador	MOVISTAR	Ecuador
<b>POR-EC</b>	Conecel S.A. (Consorcio Ecuatoriano de Telecomunicaciones S.A.)	Conecel	Ecuador
<b>ETI-EG</b>	Etisalat Misr	Etisalat	Egypt
<b>MBE-EG</b>	ECMS-Mobinil	Mobinil	Egypt
<b>VOD-EG</b>	Vodafone Egypt Telecommunications S.A.E	vodafone	Egypt
<b>CLA-SV</b>	CTE Telecom Personal SA de CV	CTE Telecom Personal SA de CV	El Salvador
<b>DIG-SV</b>	DIGICEL, S.A. de C.V.	Digicel	El Salvador
<b>FON-SV</b>	Telefonica Moviles El Salvador, S.A de c.v	Telefonica	El Salvador
<b>TMO-SV</b>	Telemovil EL Salvador S.A	TIGO	El Salvador
<b>HIT-GQ</b>	HiTs EG.SA	HiTs-GE	Equatorial Guinea
<b>ORG-GQ</b>	GETESA	Orange GQ	Equatorial Guinea
<b>ELI-EE</b>	Elisa Eesti AS	Elisa Eesti	Estonia

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>EMT-EE</b>	AS EMT	EMT GSM	Estonia
<b>TLE-EE</b>	Tele2 Eesti AS	Tele2	Estonia
<b>ETM-ET</b>	Ethiopian Telecommunications Corporation	ETMTN	Ethiopia
<b>CWF-FK</b>	Cable and Wireless Plc	Cable and Wireless	Falkland Islands (Islas Malvinas)
<b>FTG-FO</b>	Faroese Telecom	Faroese Telecom GSM	Faroe Islands
<b>VOD-FO</b>	Kall P/F	VODAFONE FO	Faroe Islands
<b>VOD-FJ</b>	Vodafone Fiji Ltd	VODAFONE	Fiji
<b>DNA-FI</b>	DNA Ltd	dna	Finland
<b>ELI-FI</b>	Elisa Corporation	Elisa Corporation	Finland
<b>FIA-FI</b>	Alands Mobiltelefon Ab	Alands Mobiltelefon AB	Finland
<b>SON-FI</b>	TeliaSonera Finland Oyj	SONERA	Finland
<b>BYT-FR</b>	Bouygues Telecom	Bouygues Telecom	France
<b>DIG-FR</b>	DIGICEL Antilles Française Guyane	DIGICEL F	France
<b>EIT-FR</b>	Euro Information Telecom	EI Telecom	France
<b>FRE-FR</b>	Free Mobile	Free Mobile	France
<b>ILI-FR</b>	Iliad	Iliad	France
<b>ORG-FR</b>	Orange France	Orange F	France

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>SFR-FR</b>	SFR	SFR	France
<b>VIN-FR</b>	Tikiphone SA	VINI	France
<b>VIR-FR</b>	Virgin Mobile	Virgin Mobile	France
<b>ONL-GF</b>	Outremer Telecom	Outremer	French Guiana
<b>LIB-GA</b>	Libertis S.A.	LIBERTIS	Gabon
<b>TLA-GA</b>	Atlantique Telecom Gabon S.A.	TELECEL	Gabon
<b>ZAI-GA</b>	Celtel Gabon SA	Zain Gabon	Gabon
<b>AFR-GM</b>	Africell (Gambia) Ltd	AFRICELL	Gambia, The
<b>GAM-GM</b>	Gambia Telecommunications Cellular Company Ltd (Gamcell)	GAMCELL	Gambia, The
<b>GMC-GM</b>	Comium Gambia Ltd	Comium Gambia	Gambia, The
<b>QC0-GM</b>	QCELL Limited	Qcell	Gambia, The
<b>BIT-GE</b>	Mobitel LLC	Mobitel	Georgia
<b>GCE-GE</b>	Geocell Ltd	GEOCELL	Georgia
<b>MAG-GE</b>	Magticom Ltd	MAGTI GSM	Georgia
<b>CON-DE</b>	Congstar	Telekom Germany	Germany
<b>EPL-DE</b>	E-Plus Mobilfunk GmbH & Co. KG	E-Plus	Germany
<b>MOB-DE</b>	Mobilcom-Debitel	Mobilcom-Debitel	Germany

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>O2O-DE</b>	Telefonica o2 Germany GmbH & Co.OHG	O2 (Germany) GmbH & Co. OHG	Germany
<b>TMO-DE</b>	T-Mobile Deutschland GmbH (DTAG)	Telekom Deutschland GmbH	Germany
<b>VOD-DE</b>	Vodafone D2 GmbH	Vodafone	Germany
<b>MTN-GH</b>	Scancom Ltd	MTN	Ghana
<b>ONE-GH</b>	Ghana Telecommunications Company Ltd	Vodafone Ghana	Ghana
<b>TIG-GH</b>	Millicom Ghana Limited	tiGO	Ghana
<b>ZAI-GH</b>	Zain Communications (Ghana) Limited	Zain	Ghana
<b>CTS-GI</b>	CTS Gibraltar Ltd	CTS Mobile	Gibraltar
<b>GIB-GI</b>	Gibtelecom Limited	GIBTEL	Gibraltar
<b>COT-GR</b>	COSMOTE - Mobile Telecommunications S.A.	COSMOTE	Greece
<b>TMO-GR</b>	T-Mobile Greece	T-Mobile	Greece
<b>TWI-GR</b>	Wind Hellas Telecommunications S.A.	WIND	Greece
<b>VOD-GR</b>	Vodafone-Panafon	vodafone	Greece
<b>TLG-GL</b>	TELE Greenland A/S	TELE Greenland	Greenland
<b>CW0-GD</b>	Cable & Wireless Grenada Ltd.	Cable & Wireless Grenada	Grenada
<b>DIG-GD</b>	Digital Grenada Ltd.	Digital	Grenada

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>DAU-GP</b>	Dauphin Telecom	DAUPHIN	Guadeloupe
<b>ORG-GP</b>	Orange Caraibe	Orange	Guadeloupe
<b>CLA-GT</b>	Telecomunicaciones De Guatemala, S.A. (TELGUA)	CLARO	Guatemala
<b>COM-GT</b>	COMCEL- Comunicaciones Celulares Sociedad Anonima	COMCEL GUATEMALA	Guatemala
<b>FON-GT</b>	Telefonica Moviles Guatemala, SA	Telefonica	Guatemala
<b>CEL-GN</b>	Cellcom Guinee S.A.	Cellcom Guinee	Guinea
<b>ORG-GN</b>	Orange Guinee SA	Orange Guinee	Guinea
<b>MTN-GW</b>	Spacetel Guinee-Bissau SA	MTN	Guinea-Bissau
<b>ORG-GW</b>	Orange Bissau	Orange Bissau	Guinea-Bissau
<b>CLN-GY</b>	Guyana Telephone & Telegraph Co.	Cellink Plus	Guyana
<b>DIG-GY</b>	U-Mobile (Cellular) Inc.	Digicel Guyana	Guyana
<b>VOI-HT</b>	Communication Cellulaire d'Haiti SA	Comcel	Haiti
<b>CLA-HN</b>	Servicios de Comunicaciones de Honduras S.A. de C.V.	CLARO GSM	Honduras
<b>FON-HN</b>	Telefonica Celular S.A (CELTEL)	CELTEL	Honduras
<b>HT2-HN</b>	Empresa Hondurena de Telecomunicaciones HONDUTEL	Empresa Hondurena de Telecomunicaciones	Honduras

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>CMH-HK</b>	CHINA MOBILE HONG KONG COMPANY LIMITED	China Mobile HK	Hong Kong SAR
<b>CSL-HK</b>	Hong Kong CSL Limited	CSL Limited	Hong Kong SAR
<b>HUT-HK</b>	Hutchison Telecom (HK) Ltd	3	Hong Kong SAR
<b>HUT-HK</b>	Hutchison Telecom (HK) Ltd	3	Hong Kong SAR
<b>PCC-HK</b>	Hong Kong Telecommunications (HKT) Limited	PCCW	Hong Kong SAR
<b>SMC-HK</b>	SmarTone Mobile Communications Limited	SmarToneVodafone	Hong Kong SAR
<b>TLH-HU</b>	Telenor Magyarorszag Zrt	Telenor 3G	Hungary
<b>TMO-HU</b>	Magyar Telekom Plc	T-Mobile Hungary	Hungary
<b>VOD-HU</b>	Vodafone Hungary Ltd	Vodafone	Hungary
<b>ICE-IS</b>	IceCell ehf	IceCell	Iceland
<b>NOV-IS</b>	Nova ehf.	NOVA	Iceland
<b>SIM-IS</b>	Siminn hf	Siminn	Iceland
<b>VIK-IS</b>	IMC Island ehf	Viking wireless	Iceland
<b>VOD-IS</b>	Og farskipti hf	Vodafone Iceland	Iceland
<b>AIR-IN</b>	Dishnet Wireless Limited	Aircel	India
<b>CLO-IN</b>	Bharat Sanchar Nigam Limited	CellOne Uttar Pradesh (East)	India

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>DOL-IN</b>	Mahanagar Telephone Nigam Ltd	Mahanagar Telephone Nigam	India
<b>ETI-IN</b>	Etisalat DB Telecom Private Limited	Etisalat DB	India
<b>IDE-IN</b>	IDEA Cellular Limited	IDEA Cellular Limited - Tamilnadu Inc Chennai	India
<b>LOO-IN</b>	Loop Telecom Pvt. Ltd.	Loop Cellular - Himachal Pradesh	India
<b>OAS-IN</b>	Bharti Hexacom Ltd	AIRTEL	India
<b>REL-IN</b>	Reliance Telecom Ltd	RELIANCE TELECOM LIMITED	India
<b>RJO-IN</b>	RJio	Reliance Jio	India
<b>TAT-IN</b>	Tata Teleservices Ltd	TATA TELESERVICES LTD	India
<b>UNI-IN</b>	Unitech Wireless Limited	uninor	India
<b>VID-IN</b>	Videocon Telecommunications Limited	Datacom Solutions Pvt. Ltd.	India
<b>VOD-IN</b>	Vodafone Essar Spacetel Limited	Vodafone NORTHEAST	India
<b>AXI-ID</b>	PT Natrindo Telepon Seluler	AXIS	Indonesia
<b>HUT-ID</b>	Hutchison CP Telecommunications	3	Indonesia
<b>IND-ID</b>	PT Indonesian Satellite Corporation Tbk (INDOSAT)	INDOSAT	Indonesia
<b>MO8-ID</b>	PT Mobile-8 Telekom Tbk	PT Mobile-8 Telekom Tbk	Indonesia

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
SFN-ID	SMARTFREN TELECOM, PT TBK	Smartfren	Indonesia
TSE-ID	PT Telekomunikasi Selular	TELKOMSEL	Indonesia
XLO-ID	Excelcom	XL	Indonesia
ASI-IQ	Asiacell Communications LLC	Asiacell	Iraq
IRA-IQ	Atheer Telecom Iraq	Zain Iraq	Iraq
KOR-IQ	Korek Telecom Ltd.	Korek Telecom	Iraq
EMB-IE	eMobile-Ireland	eMobile	Ireland
HUT-IE	Hutchison 3G Ireland limited	Hutchison 3G Ireland	Ireland
MET-IE	Meteor Mobile Telecommunications Limited	Meteor	Ireland
O2O-IE	O2 Communications (Ireland) Ltd	O2 Communications (Ireland)	Ireland
VOD-IE	Vodafone Ireland Ltd.	vodafone	Ireland
CIL-IL	Cellcom Israel Ltd	Cellcom Israel	Israel
HOT-IL	Hot Mobile Israel	Hot Mobile Israel	Israel
JAW-IL	Palestine Telecomm Co Ltd	Palestine Telecommunications Co. P.L.C	Israel
ORG-IL	Partner Communications Company Ltd	Orange	Israel
PCL-IL	Pelephone Communications Ltd.	Pelephone	Israel

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>IT3-IT</b>	H3G	H3G	Italy
<b>IWI-IT</b>	Wind Telecomunicazioni SpA	Wind Telecomunicazioni SpA	Italy
<b>OND-IT</b>	ONDA Mobile Communications	ONDA	Italy
<b>TIM-IT</b>	Telecom Italia SpA	TIM	Italy
<b>VOD-IT</b>	Vodafone Omnitel N.V.	vodafone	Italy
<b>CLA-JM</b>	Oceanic Digital Jamaica Limited	Claro Jamaica	Jamaica
<b>CW0-JM</b>	Cable & Wireless Jamaica Limited	Cable and Wireless Jamaica	Jamaica
<b>DIG-JM</b>	Digicel (Jamaica) Limited	Digicel Jamaica	Jamaica
<b>DOC-JP</b>	NTT DOCOMO, INC.	DOCOMO	Japan
<b>EML-JP</b>	eMobile, Ltd.	eMobile	Japan
<b>KDI-JP</b>	KDDI	KDDI	Japan
<b>SOF-JP</b>	SOFTBANK MOBILE Corp.	SoftBank	Japan
<b>NAV-JE</b>	Jersey Telecom	Navitas	Jersey
<b>ORG-JO</b>	Petra Jordanian Mobile Telecommunications Company (MobileCom)	Orange	Jordan
<b>UMN-JO</b>	Umniah Mobile Company	Umniah	Jordan
<b>ZAI-JO</b>	Jordan Mobile Telephone Services (JMTS)	zain JO	Jordan

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>BEE-KZ</b>	KaR-Tel LLP	Beeline	Kazakhstan
<b>KCE-KZ</b>	GSM Kazakhstan Ltd	K'CELL	Kazakhstan
<b>NEO-KZ</b>	Mobile Telecom Service LLP	Mobile Telecom Service	Kazakhstan
<b>CKL-KE</b>	Celtel Kenya Limited.	Zain Kenya	Kenya
<b>ORG-KE</b>	Telkom Kenya Limited	Orange Kenya	Kenya
<b>SAF-KE</b>	Safaricom Limited	Safaricom	Kenya
<b>KIF-KI</b>	Telecom Services Kiribati Limited (TSKL)	Kiribati Frigate	Kiribati
<b>KTO-KR</b>	KT Corporation	KT	Korea
<b>KTO-KR</b>	Korea Telecom	Korea Telecom	Korea
<b>LGU-KR</b>	LG U+	LG U+	Korea
<b>SKT-KR</b>	SK Telecom	SK Telecom	Korea
<b>VIV-KW</b>	Kuwait Telecom Company	VIVA	Kuwait
<b>WAT-KW</b>	National Mobile Telecommunications Co.	Wataniya Telecom	Kuwait
<b>ZAI-KW</b>	Mobile Telecommunications Co.	zain KW	Kuwait
<b>BEE-KG</b>	Sky Mobile LLC	Beeline KG	Kyrgyzstan
<b>MGC-KG</b>	Alfa Telecom Joint Stock Company	MEGACOM	Kyrgyzstan
<b>OOO-KG</b>	NurTelecom LLC	O!	Kyrgyzstan

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>ETL-LA</b>	Enterprise of Telecommunications Lao (ETL)	ETL MOBILE	Laos
<b>LAO-LA</b>	Lao Telecommunications	LAO TELECOMMUNICATION S	Laos
<b>TIG-LA</b>	Millicom Lao Co Ltd	Tigo	Laos
<b>UNI-LA</b>	Star Telecom Company Limited (STL)	LAT	Laos
<b>BIT-LV</b>	SIA Bite Latvija	Bite Latvija	Latvia
<b>LMT-LV</b>	Latvijas Mobilais Telefons	LMT GSM	Latvia
<b>TL2-LV</b>	TELE2	TELE2	Latvia
<b>ALF-LB</b>	MIC 1	Alfa	Lebanon
<b>MIC-LB</b>	MIC 2	MTC-Touch	Lebanon
<b>OMO-LB</b>	Ogero Telecom	Ogero Mobile (OM)	Lebanon
<b>EZI-LS</b>	Econet Telecom Lesotho (Pty) Ltd (ETL)	Econet Telecom Lesotho (Pty) Ltd	Lesotho
<b>VOC-LS</b>	Vodacom Lesotho (Pty) Ltd	VODACOM LESOTHO	Lesotho
<b>CLT-LR</b>	Cellcom Telecommunications, Inc	Cellcom Telecommunications	Liberia
<b>LON-LR</b>	Lonestar Communications Corporation	Lonestar Cell	Liberia
<b>ALM-LY</b>	AL MADAR AL JADID	MADAR	Libya
<b>LOO-LY</b>	Libyana Mobile Phone	Libyana Mobile Phone	Libya

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>FL1-LI</b>	Mobilkom (Liechtenstein) AG	FL1	Liechtenstein
<b>ORG-LI</b>	Orange (Liechtenstein) AG	Orange FL	Liechtenstein
<b>SWI-LI</b>	Swisscom (Switzerland) Ltd (Liechtenstein)	FL GSM	Liechtenstein
<b>TAN-LI</b>	TANGO Liechtenstein	Tele 2 AG	Liechtenstein
<b>BIT-LT</b>	UAB Bité Lietuva	BITE GSM	Lithuania
<b>OMT-LT</b>	OMNITEL	OMNITEL	Lithuania
<b>TL2-LT</b>	UAB TELE2	TELE2	Lithuania
<b>LUX-LU</b>	P & T Luxembourg	LUXGSM	Luxembourg
<b>ORG-LU</b>	Orange S.A.	VOXmobile	Luxembourg
<b>TAN-LU</b>	TANGO Mobile SA	TANGO	Luxembourg
<b>CTM-MO</b>	CTM	CTM	Macao SAR
<b>HUT-MO</b>	Hutchison Telephone (Macau) Company Ltd	Hutchison Telecom Macau	Macao SAR
<b>HUT-MO</b>	Hutchison Telephone (Macau) Company Ltd	Hutchison Telephone Macau	Macao SAR
<b>SMA-MO</b>	SmarTone Mobile Communications (Macau) Ltd	SmarTone	Macao SAR
<b>ONE-MK</b>	ONE Stock Company Skopje	ONE	Macedonia, FYRO
<b>TMO-MK</b>	T-Mobile Macedonia	T-Mobile Macedonia	Macedonia, FYRO

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>VIP-MK</b>	VIP OPERATOR DOOEL Skopje	VIP MKD	Macedonia, FYRO
<b>ORG-MG</b>	Orange Madagascar S.A.	Orange Madagascar	Madagascar
<b>TLM-MG</b>	Telma Mobile SA	Telma Mobile	Madagascar
<b>ZAI-MG</b>	Celtel Madagascar	ZAIN Madagascar	Madagascar
<b>CTL-MW</b>	CelTel Limited	Zain Malawi	Malawi
<b>TNM-MW</b>	Telekom Networks Malawi Ltd	TNM	Malawi
<b>CLM-MY</b>	Celcom (Malaysia) Sdn Bhd	CELCOM GSM	Malaysia
<b>DIG-MY</b>	DiGi Telecommunications Sdn Bhd	DiGi	Malaysia
<b>MYM-MY</b>	Maxis Communications Berhad	MMS & MB	Malaysia
<b>UMO-MY</b>	U Mobile Sdn. Bhd.	U Mobile	Malaysia
<b>DMO-MV</b>	Dhivehi Raajjeypge Gulhun Private Ltd	Dhiraagu	Maldives
<b>WAT-MV</b>	Wataniya Telecom Maldives Pvt. Ltd	Wataniya Telecom Maldives Pvt. Ltd	Maldives
<b>MAL-ML</b>	Malitel SA	MALITEL	Mali
<b>ORG-ML</b>	Orange Mali SA	Orange MALI	Mali
<b>GOM-MT</b>	Mobisle Communications Limited	go mobile	Malta
<b>MEM-MT</b>	Melita Mobile Ltd.	3 G Telecoms Malta	Malta

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>VOD-MT</b>	Vodafone Malta Limited	vodafone	Malta
<b>EMR-MR</b>	CHINGUITEL S.A.	Chinguitel	Mauritania
<b>MAT-MR</b>	MATTEL	MATTEL	Mauritania
<b>IUS-MX</b>	Iusacell S.A. de C.V.	Iusacell S.A. de C.V.	Mexico
<b>MOV-MX</b>	Telefonica Moviles Mexico	movistar	Mexico
<b>NXT-MX</b>	NEXTEL	Nextel	Mexico
<b>TLC-MX</b>	Radiomovil Dipsa SA de CV (TELCEL)	TELCEL GSM	Mexico
<b>UNE-MX</b>	Unefon	Iusacell (Iusacell S.A. de C.V.)	Mexico
<b>FSM-FM</b>	FSM Telecommunications Corporation	FSM Telecommunications Corporation	Micronesia
<b>MDC-MD</b>	Moldcell SA	MOLDECELL	Moldova
<b>ORG-MD</b>	Orange Moldova S.A.	VoXtel	Moldova
<b>UNI-MD</b>	MOLDTELECOM	UNITE	Moldova
<b>MON-MC</b>	Monaco Telecom	Monaco Telecom	Monaco
<b>MOB-MN</b>	MobiCom	MobiCom	Mongolia
<b>UNT-MN</b>	Unitel LLC	Unitel	Mongolia
<b>TMO-ME</b>	T-Mobile Montenegro	T-Mobile Montenegro	Montenegro
<b>TOR-ME</b>	Telenor Ltd Montenegro	Telenor Montenegro	Montenegro

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>MED-MA</b>	Méditel	Méditel	Morocco
<b>MRT-MA</b>	Maroc Telecom	Maroc Telecom	Morocco
<b>WAN-MA</b>	Wana	Wana	Morocco
<b>MCE-MZ</b>	Mocambique Celular S.A.R.L (mcel)	mcel	Mozambique
<b>VOC-MZ</b>	VM, S.A.R.L.	Vodacom Mozambique	Mozambique
<b>MPT-MM</b>	Myanmar Posts and Telecommunications	MPT GSM Network	Myanmar
<b>000-11</b>	Open market devices (North America)		N/A
<b>000-22</b>	Open market devices (Latin America)		N/A
<b>000-23</b>	Open market devices (North Latin America)		N/A
<b>000-24</b>	Open market devices (South Latin America)		N/A
<b>000-33</b>	Open market devices (Europe)		N/A
<b>000-34</b>	Open market devices (Eastern Europe)		N/A
<b>000-35</b>	Open market devices (Western Europe)		N/A
<b>000-36</b>	Open market devices (Northern Europe)		N/A
<b>000-37</b>	Open market devices (Southern Europe)		N/A

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>000-44</b>	Open market devices (Africa)		N/A
<b>000-55</b>	Open market devices (Middle East)		N/A
<b>000-66</b>	Open market devices (Asia)		N/A
<b>000-67</b>	Open market devices (South East Asia)		N/A
<b>000-68</b>	Open market devices (Central Asia)		N/A
<b>000-77</b>	Open market devices (Oceania)		N/A
<b>000-88</b>	Open market devices (World Wide)		N/A
<b>LEO-NA</b>	Powercom (Pty) Ltd	leo™	Namibia
<b>MTC-NA</b>	MTC Namibia	MTC	Namibia
<b>NCE-NP</b>	Spice Nepal Private Ltd	Ncell	Nepal
<b>BEN-NL</b>	BEN	T-Mobile Netherlands	Netherlands
<b>NLK-NL</b>	KPN B.V.	KPN B.V.	Netherlands
<b>T2L-NL</b>	Tele2 Netherlands	Tele2 Netherlands	Netherlands
<b>TMO-NL</b>	T-Mobile Netherlands B.V	T-Mobile NL	Netherlands
<b>VOD-NL</b>	Vodafone Libertel B.V.	vodafone	Netherlands
<b>CHI-AN</b>	Setel NV	UTS Wireless Curacao	Netherlands Antilles (former)

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>CTG-AN</b>	Curacao Telecom N.V.	Digicel Netherlands Antilles	Netherlands Antilles (former)
<b>TEV-AN</b>	Telcell N.V.	Telcell N.V.	Netherlands Antilles (former)
<b>MMC-NC</b>	OPT New Caledonia	MOBILIS	New Caledonia
<b>DE2-NZ</b>	Two Degrees Mobile Limited	2degrees	New Zealand
<b>TEL-NZ</b>	Telecom New Zealand Limited	Telecom New Zealand	New Zealand
<b>VOD-NZ</b>	Vodafone Mobile NZ Limited	vodafone	New Zealand
<b>CLA-NI</b>	Empresa Nicaraguense de Telecomunicaciones S.A. - ENITEL	CLARO NIC	Nicaragua
<b>MOV-NI</b>	Telefonia Celular de Nicaragua S.A.	movistarNI	Nicaragua
<b>CEL-NE</b>	Celtel Niger	Zain Niger	Niger
<b>ORG-NE</b>	Orange Niger S.A	Orange Niger	Niger
<b>TNI-NE</b>	Telecel Niger SA	Telecel Niger	Niger
<b>AIR-NG</b>	Airtel Nigeria	Airtel Nigeria	Nigeria
<b>EMT-NG</b>	Emerging Markets Telecommunication Services Ltd	EMTS	Nigeria
<b>GLO-NG</b>	Globacom Ltd.	Glo Mobile	Nigeria
<b>MTN-NG</b>	MTN Nigeria Communications Limited	MTN Nigeria	Nigeria
<b>ZAI-NG</b>	Celtel Nigeria Ltd	Celtel Nigeria Ltd (Zain)	Nigeria

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>NT0-NF</b>	Norfolk Telecom	Norfolk Telecom	Norfolk Island
<b>MBN-NO</b>	Mobile Norway AS	Mobile Norway	Norway
<b>NET-NO</b>	NETCOM AS	NetCom	Norway
<b>TNO-NO</b>	Telenor Norge AS.	TELENOR	Norway
<b>NAW-OM</b>	Omani Qatari Telecommunications Company SAOC	nawras	Oman
<b>OMA-OM</b>	Oman Telecommunications Company	OMAN MOBILE	Oman
<b>MIB-PK</b>	Mobilink-PMCL	Mobilink	Pakistan
<b>TLN-PK</b>	Telenor Pakistan (Pvt) Ltd.	Telenor Pakistan (Pvt) Ltd.	Pakistan
<b>UFO-PK</b>	Pakistan Telecommunication Mobile Ltd	Ufone	Pakistan
<b>WAR-PK</b>	Warid Telecom (PVT) Ltd	Warid Telecom	Pakistan
<b>ZON-PK</b>	CMPak Limited	ZONG	Pakistan
<b>PLW-PW</b>	Palau Mobile Corporation	Palau Mobile	Palau
<b>WM0-PS</b>	Wataniya Palestine Mobile Telecommunications Company	Wataniya Mobile	Palestinian Authority
<b>CLA-PA</b>	Claro Panama S.A.	Claro Panama	Panama
<b>FON-PA</b>	TELEFONICA MOVILES PANAMA, S.A.	Movistar	Panama

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>PAN-PA</b>	Cable & Wireless Panama	Cable & Wireless Panama	Panama
<b>BMO-PG</b>	BMobile Limited	Bee Mobile	Papua New Guinea
<b>DIG-PG</b>	Digicel PNG Ltd.	Digicel PNG	Papua New Guinea
<b>CLA-PY</b>	AMX Paraguay S.A.	CLARO PARAGUAY	Paraguay
<b>FON-PY</b>	Telefonica Celular Del Paraguay S.A. (Telecel S.A.)	Telecel Paraguay	Paraguay
<b>HPG-PY</b>	Hola Paraguay S.A.	VOX	Paraguay
<b>PER-PY</b>	Nucleo S.A	Personal	Paraguay
<b>FON-PE</b>	Telefonica Moviles S.A.	Movistar Peru	Peru
<b>MOV-PE</b>	America Movil Peru S.A.C	CLARO PER	Peru
<b>NXT-PE</b>	Entel Peru S.A.	Entel Peru	Peru
<b>VTL-PE</b>	Viettel Peru	Viettel Peru	Peru
<b>GLO-PH</b>	Globe Telecom	Globe Telecom	Philippines
<b>ISL-PH</b>	Innove Communications, Inc.	Islacom	Philippines
<b>SMA-PH</b>	Smart Communications Inc	SMART Gold	Philippines
<b>SUN-PH</b>	Digital Telecommunications Phils, Inc	Digitel Mobile/Sun Cellular	Philippines
<b>ERA-PL</b>	Polska Telefonia Cyfrowa (T-Mobile)	Era	Poland

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>HYH-PL</b>	Heyah	T-Mobile PL	Poland
<b>ORG-PL</b>	PTK Centertel	Orange	Poland
<b>PLA-PL</b>	P4 Sp. z o.o	P4	Poland
<b>PLU-PL</b>	Polkomtel S.A.	Plus	Poland
<b>TMO-PL</b>	T-Mobile Poland	T-Mobile	Poland
<b>MEO-PT</b>	Meo	Meo	Portugal
<b>OPT-PT</b>	Sonaecom - Serviços de Comunicações, S.A.	OPTIMUS	Portugal
<b>TMN-PT</b>	TMN	TMN	Portugal
<b>VOD-PT</b>	Vodafone Portugal	vodafone	Portugal
<b>CLA-PR</b>	Puerto Rico Telephone Company Inc.	Claro GSM	Puerto Rico
<b>QTE-QA</b>	Q-Tel	Qatarnet	Qatar
<b>VFQ-QA</b>	Vodafone Qatar Q.S.C.	Vodafone Qatar	Qatar
<b>ORG-RE</b>	Orange Reunion	Orange reunion	Reunion
<b>OUT-RE</b>	OUTREMER TELECOM	OUTREMER TELECOM	Reunion
<b>SFR-RE</b>	Societe Reunionnaise de Radiotelephone	SFR REUNION	Reunion
<b>COS-RO</b>	Cosmote Romanian Mobile Telecommunications S.A.	COSMOTE	Romania
<b>DIG-RO</b>	RCS & RDS	Digi.Mobil	Romania

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>ORG-RO</b>	Orange Romania SA	ORANGE	Romania
<b>TMO-RO</b>	Telekom Romania	Telekom	Mexico
<b>VFR-RO</b>	Vodafone Romania S.A.	Vodafone Romania S.A.	Romania
<b>AKO-RU</b>	AKOS CJSC	AKOS	Russia
<b>BEE-RU</b>	OJSC VimpelCom	Beeline	Russia
<b>BMT-RU</b>	BM Telecom Ltd.	BM Telecom	Russia
<b>BWC-RU</b>	Baykalwestcom	Baykalwestcom	Russia
<b>DTC-RU</b>	DonTeleCom	DTC	Russia
<b>ETK-RU</b>	Yeniseytelecom	Yeniseitelecom	Russia
<b>IND-RU</b>	CJSC Volgograd Mobile	INDIGO	Russia
<b>KOD-RU</b>	ZAO Kodotel	KODOTEL	Russia
<b>KUG-RU</b>	Kuban- GSM Closed JSC	Kuban-GSM	Russia
<b>MEG-RU</b>	MegaFon, Open Joint Stock Company	Megafon	Russia
<b>MOT-RU</b>	LLC Ekaterinburg-2000	MOTIV	Russia
<b>MTS-RU</b>	Mobile TeleSystems (MTS)	MTS	Russia
<b>NTC-RU</b>	New Telephone Company	New Telephone Company	Russia
<b>RUS-RU</b>	SIBCHALLENGE LTD	SIBCHALLENGE	Russia
<b>SMA-RU</b>	PJSC SMARTS	SMARTS	Russia

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>TL2-RU</b>	St. Petersburg Telecom	TELE2	Russia
<b>URA-RU</b>	Uraltel	Uraltel	Russia
<b>UTE-RU</b>	JSC Uralsvyazinform	Utel (RUS17)	Russia
<b>VPC-RU</b>	Vimpelcom	Vimpelcom	Russia
<b>RCE-RW</b>	MTN Rwandacell SARL	MTN-RWA	Rwanda
<b>RWT-RW</b>	RWANDATEL S.A.	RWANDATEL	Rwanda
<b>TIG-RW</b>	TIGO RWANDA S.A	TIGO RWANDA	Rwanda
<b>LIM-KN</b>	Cable & Wireless St Kitts & Nevis Limited	Cable & Wireless St Kitts & Nevis	Saint Kitts and Nevis
<b>CW0-LC</b>	Cable & Wireless Caribbean Cellular (St Lucia) Limited	Cable & Wireless	Saint Lucia
<b>DIG-LC</b>	Digicel (St Lucia) Limited	Digicel (St Lucia)	Saint Lucia
<b>AME-PM</b>	SPM Telecom	AMERIS	Saint Pierre and Miquelon
<b>CW0-VC</b>	Cable & Wireless Caribbean Cellular (St. Vincent & the Grenadines) Ltd	Cable & Wireless Caribbean Cellular (St. Vincent & the Grenadines)	Saint Vincent and the Grenadines
<b>DIG-VC</b>	Digicel (St. Vincent and the Grenadines) Limited	Digicel (St. Vincent and the Grenadines)	Saint Vincent and the Grenadines
<b>GO0-WS</b>	Samoatel Limited	Samoatel Mobile	Samoa
<b>SMT-SM</b>	San Marino Telecom Spa	San Marino Telecom	San Marino
<b>CST-ST</b>	Companhia Santomense de Telecomunicacoes SARL	CSTmovel	São Tomé and Príncipe

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>MBL-SA</b>	Etihad Etisalat Company	Mobily	Saudi Arabia
<b>STC-SA</b>	Saudi Telecom Company (STC)	STC	Saudi Arabia
<b>ZAI-SA</b>	Zain	Zain	Saudi Arabia
<b>ALI-SN</b>	Sonatel	Alize	Senegal
<b>EXP-SN</b>	Sudan Telecom Company Ltd	Expresso Senegal	Senegal
<b>SEN-SN</b>	SENTEL GSM S.A.	SENTEL	Senegal
<b>TEL-RS</b>	Telekom Srbija	Telekom Srbija	Serbia
<b>TOR-RS</b>	Telenor d.o.o Serbia	Telenor Serbia	Serbia
<b>PRO-CS</b>	Telenor D.o.o. Podgorica	Telenor	Serbia and Montenegro (former)
<b>SCG-CS</b>	Telekom Srbija	Telekom Srbija	Serbia and Montenegro (former)
<b>VIP-CS</b>	Vip mobile d.o.o.	Vip	Serbia and Montenegro (former)
<b>CWS-SC</b>	Cable & Wireless (Seychelles)	CABLE & WIRELESS	Seychelles
<b>AFR-SL</b>	LINTEL (Sierra Leone) Limited	Africell	Sierra Leone
<b>CEL-SL</b>	Celtel (SL) Limited	Zain Sierra Leone	Sierra Leone
<b>GRE-SL</b>	Ambitel (Sierra Leone) Limited	GreenN™ Sierra Leone	Sierra Leone
<b>SLE-SL</b>	Comium Sierra Leone INC	COMIUM Sierra Leone	Sierra Leone

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>M13-SG</b>	M1 Limited	M1	Singapore
<b>SIN-SG</b>	Singapore Telecom Mobile Pte Ltd	SingTel	Singapore
<b>STA-SG</b>	StarHub Mobile Pte Ltd	StarHub	Singapore
<b>O2O-SK</b>	Telefónica O2 Slovakia, s.r.o.	O2 - SK	Slovakia
<b>ORG-SK</b>	Orange Slovensko a.s.	Orange SK	Slovakia
<b>TMO-SK</b>	Slovak Telekom, a. s.	T-Mobile SK	Slovakia
<b>JAN-SI</b>	Janustrade Slovenia	Janustrade	Slovenia
<b>MBT-SI</b>	Mobitel D.D.	MOBITEL	Slovenia
<b>SIM-SI</b>	SI.MOBIL d.d.	SI.mobil d.d	Slovenia
<b>T20-SI</b>	T-2 d.o.o.	T-2	Slovenia
<b>TUS-SI</b>	Tusmobil d.o.o.	Tusmobil	Slovenia
<b>BRE-SB</b>	Solomon Telekom Co Ltd	BREEZE	Solomon Islands
<b>GOL-SO</b>	Golis Telecommunications Company Ltd	Golis	Somalia
<b>MNY-SO</b>	Montysom LTD	Montysom LTD	Somalia
<b>SOM-SO</b>	Somtel GSM FZCO (Somtel International Ltd.)	SOMTEL	Somalia
<b>TLS-SO</b>	Telesom Company	Telesom	Somalia
<b>CSA-ZA</b>	Cell C (Pty) Ltd	Cell C	South Africa

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>MTN-ZA</b>	MTN - Mobile Telephone Networks (Pty) Ltd.	MTN - Mobile Telephone Network	South Africa
<b>VOD-ZA</b>	Vodacom Group Pty Ltd.	VodaCom	South Africa
<b>FON-ES</b>	Telefonica Moviles Espana S.A.	movistar	Spain
<b>JAZ-ES</b>	Jazztel	Jazztel	Spain
<b>ORG-ES</b>	France Telecom España SA	Orange	Spain
<b>VOD-ES</b>	Vodafone Espana S.A.U.	vodafone	Spain
<b>YOI-ES</b>	Xfera Moviles SA	Yoigo	Spain
<b>HUT-LK</b>	Hutchison Telecommunications Lanka (Pte) Limited	Hutchison Telecommunications Lanka (Pvt)	Sri Lanka
<b>DIG-SR</b>	Digicel Suriname NV	Digicel Suriname NV	Suriname
<b>TSG-SR</b>	Telesur	TELESUR.GSM	Suriname
<b>UNI-SR</b>	Intelsur N.V.	Intelsur N.V.	Suriname
<b>SWA-SZ</b>	Swazi MTN Limited	Swazi MTN	Swaziland
<b>DJU-SE</b>	Djuice	Telenor	Sweden
<b>DKT-SE</b>	TDC Mobil	TDC	Sweden
<b>MBI-SE</b>	MobiSir	Trafikverket	Sweden
<b>SE3-SE</b>	Hi3G Access AB	3	Sweden
<b>SWG-SE</b>	Sweden2G	Net4Mobility	Sweden

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>TIA-SE</b>	TeliaSonera Mobile Networks AB Sweden	TeliaSonera Mobile Networks	Sweden
<b>TL2-SE</b>	Tele2	Tele2	Sweden
<b>TOR-SE</b>	Telenor	Telenor	Sweden
<b>VEN-SE</b>	Ventelo	Ventelo Sverige	Sweden
<b>INP-CH</b>	In &Phone SA	in&phone	Switzerland
<b>ONA-CH</b>	OnAir Switzerland Sarl	ONAIR	Switzerland
<b>ORG-CH</b>	Orange Communications SA	ORANGE	Switzerland
<b>SUN-CH</b>	Sunrise Communications AG	Sunrise	Switzerland
<b>SWI-CH</b>	Swisscom (Switzerland) Ltd	Swisscom	Switzerland
<b>CHT-TW</b>	Chunghwa Telecom	Chunghwa Telecom	Taiwan
<b>FET-TW</b>	Far EasTone Telecommunications Co Ltd	Far EasTone	Taiwan
<b>TWM-TW</b>	Taiwan Mobile Co.Ltd	Taiwan Mobile	Taiwan
<b>VIB-TW</b>	VIBO Telecom Inc	VIBO	Taiwan
<b>BAB-TJ</b>	CJSC Babilon-Mobile	Babilon-M	Tajikistan
<b>BEE-TJ</b>	Tacom LLC	BEELINE TJ	Tajikistan
<b>MLT-TJ</b>	TT Mobile, Closed joint-stock company	Mobile Lines of Tajikistan	Tajikistan
<b>TCE-TJ</b>	JV Somoncom	JV Somoncom	Tajikistan

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>CEL-TZ</b>	Celtel Tanzania Limited	Zain Tanzania.	Tanzania
<b>HIT-TZ</b>	Excellentcom Tanzania Ltd	Hits Tanzania	Tanzania
<b>TIG-TZ</b>	MIC Tanzania Limited	Tigo	Tanzania
<b>VOC-TZ</b>	Vodacom Tanzania Limited	Vodacom Tanzania Limited	Tanzania
<b>ZAN-TZ</b>	Zanzibar Telecom Ltd	ZANTEL	Tanzania
<b>ACT-TH</b>	TOT PLC	TOT Mobile	Thailand
<b>ARS-TH</b>	ACeS Regional Services Co.,Ltd. (ARS)	ACes Sat Phone Thailand	Thailand
<b>AWN-TH</b>	Advanced Wireless Network 1	AWN Thailand (AIS 3G)	Thailand
<b>CAT-TH</b>	CAT Telecom Public Company Limited	CAT Telecom Thailand	Thailand
<b>DTA-TH</b>	Total Access Communications Co	DTAC	Thailand
<b>GSM-TH</b>	Digital Phone Co Ltd	GSM 1800	Thailand
<b>THG-TH</b>	Advanced Info Service PLC	AIS GSM	Thailand
<b>TRU-TH</b>	True Move Company Ltd	True Move	Thailand
<b>TT0-TL</b>	Timor Telecom	Timor Telecom	Timor-Leste
<b>TGC-TG</b>	Togo Telecom	TOGO CELL	Togo
<b>TTG-TG</b>	Telecel Togo	Telecel Togo	Togo
<b>DIG-TO</b>	Digicel (Tonga) Limited	Digicel (Tonga) Limited	Tonga

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>UCA-TO</b>	Tonga Communications Corporation	U-CALL	Tonga
<b>TST-TT</b>	Telecommunications Services of Trinidad and Tobago Ltd	TSTT	Trinidad and Tobago
<b>ORG-TN</b>	ORANGE TUNISIE, SA	Orange Tunisie	Tunisia
<b>TUN-TN</b>	Tunisie Telecom	TUNTEL	Tunisia
<b>AVE-TR</b>	Avea Iletisim Hizmetleri A.S.	AVEA	Turkey
<b>TCE-TR</b>	Turkcell Iletisim Hizmetleri A.S.	Turkcell Iletisim Hizmetleri	Turkey
<b>VFT-TR</b>	Vodafone Telekomunikasyon A.S	Vodafone Turkey	Turkey
<b>MTS-TM</b>	Barash Communication Technologies INC	MTS Turkmenistan	Turkmenistan
<b>TMC-TM</b>	Altyn Asyr MC	Altyn Asyr	Turkmenistan
<b>CWO-TC</b>	Cable & Wireless West Indies Ltd (Turks & Caicos)	Cable & Wireless West Indies (Turks & Caicos)	Turks and Caicos Islands
<b>ICO-TC</b>	Islandcom Telecommunications	Islandcom	Turks and Caicos Islands
<b>CEL-UG</b>	Celtel Uganda Limited	Zain Uganda	Uganda
<b>MTN-UG</b>	MTN Uganda Ltd	MTN-UGANDA	Uganda
<b>ORG-UG</b>	Orange Uganda Limited	ORANGE UGANDA LIMITED	Uganda
<b>UTL-UG</b>	Uganda Telecom Ltd	Uganda Telecom Ltd	Uganda

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>WAR-UG</b>	Warid Telecom Uganda Limited	Warid Telecom Uganda	Uganda
<b>BEE-UA</b>	Ukrainian Radio Systems	Beeline UA	Ukraine
<b>GT0-UA</b>	Golden Telecom LLC	GOLDEN TELECOM	Ukraine
<b>LIF-UA</b>	Astelit LLC	life:)	Ukraine
<b>MTS-UA</b>	MTS Ukraine	MTS UKR	Ukraine
<b>UAK-UA</b>	Kyivstar GSM JSC	KYIVSTAR	Ukraine
<b>DU0-AE</b>	Emirates Integrated Telecommunications Company PJSC	du	United Arab Emirates
<b>ETI-AE</b>	Emirates Telecom Corp- ETISALAT	Emirates Telecom Corp- ETISALAT	United Arab Emirates
<b>BRT-GB</b>	British Telecom (EE MVNO)	British Telecom	United Kingdom
<b>CWU-GB</b>	Cable and Wireless UK	Cable and Wireless UK (England)	United Kingdom
<b>GAF-GB</b>	GiffGaff UK Network	GiffGaff UK Network	United Kingdom
<b>HUT-GB</b>	Hutchison 3G UK Ltd	3	United Kingdom
<b>JTW-GB</b>	Wave Telecom	JT-Wave	United Kingdom
<b>LFE-GB</b>	LIFE Phones4U MVNO	LIFE	United Kingdom
<b>MAN-GB</b>	Manx Telecom	Manx	United Kingdom
<b>O2O-GB</b>	Telefónica O2 UK Limited	O2 (UK)	United Kingdom

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>ORG-GB</b>	Everything Everywhere Limited	Orange	United Kingdom
<b>PMN-GB</b>	TeleWare PLC	PMN	United Kingdom
<b>SUR-GB</b>	Cable & Wireless Jersey Limited	Cable & Wireless Jersey Ltd (Sure)	United Kingdom
<b>TES-GB</b>	Tesco Mobile UK	Tesco Mobile UK	United Kingdom
<b>TMO-GB</b>	Everything Everywhere Limited	T-Mobile UK	United Kingdom
<b>UK0-GB</b>	Mundio Mobile Limited	MCom	United Kingdom
<b>VOD-GB</b>	VODAFONE Ltd	vodafone	United Kingdom
<b>000-00</b>	Microsoft internal use		United States
<b>ACG-US</b>	Associated Carrier Group	Associated Carrier Group	United States
<b>ACS-US</b>	Advantage Cellular Systems Inc	Advantage	United States
<b>AIO-US</b>	Cricket Wireless (AT&T MVNO)	Cricket Wireless	United States
<b>ALT-US</b>	Alltel Wireless	Alltel	United States
<b>ARD-US</b>	Airadigm Communications	Airadigm Communications	United States
<b>AST-US</b>	Arctic Slope Telephone Association Cooperative	Arctic Slope Telephone Association Cooperative	United States
<b>ATT-US</b>	AT&T Mobility	AT&T	United States
<b>BRT-US</b>	Brightspot Mobile	Brightspot Mobile	United States

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>BST-US</b>	Boost	Sprint PCS	United States
<b>BTW-US</b>	Michigan Wireless, LLC	Bug Tussel Wireless	United States
<b>C1E-US</b>	Cellular Properties, Inc	Cellular One of East Central Illinois	United States
<b>CBW-US</b>	Cincinnati Bell Wireless	Cincinnati Bell Wireless	United States
<b>CEL-US</b>	Cellcom	Verizon MVNO	United States
<b>CEN-US</b>	Centennial Communications	Centennial Communications	United States
<b>CHI-US</b>	MTPCS, LLC	Cellular One	United States
<b>CLS-US</b>	Cellular South	Cellular South	United States
<b>COM-US</b>	Commnet Wireless, LLC	Commnet	United States
<b>CON-US</b>	Consumer Cellular	T-Mobile MVNO	United States
<b>COR-US</b>	Corr Wireless Communications	Corr Wireless Communications	United States
<b>CRK-US</b>	Cricket Wireless	Cricket Wireless	United States
<b>CTX-US</b>	TX-11 Acquisition, LLC	Cellular One of East Texas	United States
<b>CWC-US</b>	Cordova Wireless Communications Inc	Cordova Wireless Communications	United States
<b>CWL-US</b>	Choice Wireless LC	CellularOne of Texoma	United States
<b>DHA-US</b>	Alaska Wireless Communications, LLC	Dutch Harbor	United States
<b>DOC-US</b>	DOCOMO PACIFIC, INC.	DOCOMO PACIFIC, INC	United States

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>FAM-US</b>	Family Mobile	T-Mobile MVNO	United States
<b>GCI-US</b>	GCI Communications Corp.	GCI Communications Corp.	United States
<b>ICA-US</b>	Wave Runner LLC Mariana Islands	i CAN_GSM	United States
<b>IDT-US</b>	IDT/TuYo Mobile	T-Mobile MVNO	United States
<b>IMM-US</b>	Keystone Wireless LLC	Immix Wireless	United States
<b>IND-US</b>	Indigo Wireless, Inc	Indigo Wireless	United States
<b>ISM-US</b>	iSmart Mobile, LLC	Big Sky Mobile	United States
<b>ITE-US</b>	IT&E Overseas, Inc	IT&E Wireless	United States
<b>IWS-US</b>	Iowa Wireless Services, LLC	Iowa Wireless Services, LLC	United States
<b>JAS-US</b>	Jasper Wireless, inc	Jasper	United States
<b>LAM-US</b>	Lamar County Cellular, Inc	Lamar County Cellular	United States
<b>LPW-US</b>	Leap Wireless	Leap Wireless	United States
<b>LRA-US</b>	LTE Rural America	LTE Rural America	United States
<b>LYC-US</b>	Lyca Mobile	T-Mobile MVNO	United States
<b>MAI-US</b>	Maine PCS, LLC.	Maine PCS	United States
<b>MID-US</b>	Mid-Tex Cellular, Ltd.	Mid-Tex Cellular	United States
<b>MOS-US</b>	CTC Telcom, Inc.	CTC Telcom Wireless	United States
<b>MPL-US</b>	Pulse Mobile LLC	GTA Wireless	United States

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>NCW-US</b>	NewCore Wireless, LLC	NewCore Wireless	United States
<b>NEC-US</b>	North East Colorado Cellular, Inc (NECCI)	Viaero Wireless	United States
<b>NET-US</b>	Net10	T-Mobile MVNO	United States
<b>NYP-US</b>	AT&T Mobility	AT&T	United States
<b>OTZ-US</b>	OTZ Telecommunications Inc.	OTZ Cellular	United States
<b>PAC-US</b>	Kaplan Telephone Company,Inc	PACE	United States
<b>PCS-US</b>	Metro PCS	Metro PCS	United States
<b>PET-US</b>	Petrocom LLC	PetroCom	United States
<b>PIN-US</b>	Pine Telephone Company	Pine Cellular	United States
<b>PLA-US</b>	Texas RSA 3 Ltd Partnership	Plateau Wireless	United States
<b>PRO-US</b>	Proximiti Mobility, Inc	Proximiti Mobility	United States
<b>RED-US</b>	Red Pocket	T-Mobile MVNO	United States
<b>ROA-US</b>	Roam Mobility	T-Mobile MVNO	United States
<b>SIM-US</b>	TMP Corp	SIMMETRY	United States
<b>SMA-US</b>	Go Smart	T-Mobile MVNO	United States
<b>SOL-US</b>	Solavei	T-Mobile MVNO	United States
<b>SPM-US</b>	Simple Mobile	T-Mobile MVNO	United States

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>SPP-US</b>	Sprint Prepaid	Sprint Prepaid	United States
<b>SPT-US</b>	Sprint	Sprint PCS	United States
<b>SPW-US</b>	Sprint Wholesale	Sprint PCS	United States
<b>STE-US</b>	Stelera Wireless, L.L.C.	Stelera Wireless	United States
<b>SXL-US</b>	Long Lines Wireless LLC	Long Lines Wireless	United States
<b>TMO-US</b>	T-Mobile USA, Inc	T-Mobile USA	United States
<b>TRF-US</b>	TracFone	TracFone	United States
<b>UNI-US</b>	Union Telephone Company	Union Telephone Company	United States
<b>UNV-US</b>	Univision Mobile	T-Mobile MVNO	United States
<b>USC-US</b>	United States Cellular Corporation	U.S.Cellular	United States
<b>VIR-US</b>	Virgin Mobile	Sprint PCS	United States
<b>VZW-US</b>	Verizon Wireless	Wireless Alliance	United States
<b>WES-US</b>	WestLink Communications LLC	Westlink Communications	United States
<b>WIL-US</b>	Wilkes Cellular Inc	Via Wireless	United States
<b>XIT-US</b>	XIT Wireless	Texas RSA 1 Ltd Partnership DBA XIT Cell	United States
<b>YOR-US</b>	Yorkville Telephone Cooperative	Yorkville Telephone Cooperative	United States
<b>ANC-UY</b>	Ancel	antel Telefonía móvil	Uruguay

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>CLA-UY</b>	AM Wireless Uruguay S.A.	CLARO URUGUAY	Uruguay
<b>FON-UY</b>	Telefónica Móviles Uruguay	Moviestar	Uruguay
<b>BEE-UZ</b>	Unitel LLC	Beeline	Uzbekistan
<b>UCE-UZ</b>	Coscom	Ucell	Uzbekistan
<b>UZB-UZ</b>	FE 'Uzdunrobita' Ltd	Uzdunrobita GSM	Uzbekistan
<b>DIG-VU</b>	Digicel (Vanuatu) Ltd	Digicel	Vanuatu
<b>SMI-VU</b>	TELECOM VANUATU LIMITED	SMILE	Vanuatu
<b>DIG-VE</b>	Corporacion Digitel C.A.	DIGITEL GSM	Venezuela
<b>FON-VE</b>	Telefonica Moviles Venezuela S.A. (Telcel)	movistar	Venezuela
<b>MOV-VE</b>	Telecomunicaciones Movilnet C.A.	Telecomunicaciones Movilnet	Venezuela
<b>GTL-VN</b>	GTEL Mobile Joint Stock Company (GTEL Mobile)	GTEL Mobile	Vietnam
<b>VIE-VN</b>	Viettel Corporation	Viettel Mobile	Vietnam
<b>VMO-VN</b>	Vietnamobile	Vietnamobile	Vietnam
<b>VMS-VN</b>	Vietnam Mobile Telecom Services Company	Mobifone	Vietnam
<b>VNM-VN</b>	Vietnam Telecom Services Company	Vinaphone	Vietnam
<b>CCT-VG</b>	Caribbean Cellular Telephone	CCT	British Virgin Islands

MOBILE OPERATOR ID	ORGANIZATION	NETWORK	COUNTRY/REGION
<b>CW0-VG</b>	Cable & Wireless (West Indies) Limited	Cable & Wireless (West Indies)	British Virgin Islands
<b>MTN-YE</b>	Spacetel - Yemen	MTN	Yemen
<b>SAB-YE</b>	Yemen Mobile Phone Company - Sabafon	Yemen Company for Mobile Telephony	Yemen
<b>Y00-YE</b>	HiTS-UNITEL	Y	Yemen
<b>CZL-ZM</b>	Celtel Zambia Limited	Zain Zambia	Zambia
<b>MTN-ZM</b>	MTN (Zambia) Ltd	MTN ZAMBIA	Zambia
<b>ECO-ZW</b>	Econet Wireless (Private) Limited	ECONET	Zimbabwe
<b>NET-ZW</b>	NetOne Cellular ( <i>Pvt Ltd</i> )	NetOne Cellular	Zimbabwe
<b>TLZ-ZW</b>	Telecel Zimbabwe (PVT) Ltd	TELECEL	Zimbabwe

## Requests for a new MOID

Partners that need a new mobile operator ID can request one by contacting their Microsoft representative. As part of the request, partners must provide the following information:

- The business justification and impact for your request.
- The regions or mobile operator for whom you would like the new MOID generated for. Provide the mobile operator's name, network, and country/region.
- Specify if the requested MOID is for a MVNO.
- Link to the mobile operator's Web site so Microsoft can confirm the name and location.
- If there is a bug or work item ID to track this request, provide the ID.

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Registry values for carrier-unlocked phones

10/2/2018 • 4 minutes to read • [Edit Online](#)

Values to use for the mobile operator registry setting.

The following table contains the values to use for the mobile operator registry setting

**PhoneMobileOperatorName** when building carrier-unlocked phones. For more information about the registry key, see [Phone metadata in DeviceTargetingInfo](#). If you know the mobile operator, use the [Registry values for mobile operator IDs](#) instead.

The following table shows the mobile operator ID values for phones that will be sold in a large geographic area instead of a single country/region.

## Note

Note that it is not possible to prevent an update from going to one or more countries/regions in that geographical area. If any of the following regional IDs is specified, updates can either go to the entire region or none of it. There is no way to update phones with more granularity.

MOBILE OPERATOR ID	REGION
000-11	North America
000-22	Latin America
000-23	North Latin America
000-24	South Latin America
000-33	Europe
000-34	Eastern Europe
000-35	Western Europe
000-36	Northern Europe
000-37	Southern Europe
000-44	Africa
000-55	Middle East

MOBILE OPERATOR ID	REGION
000-66	Asia
000-67	South East Asia
000-68	Central Asia
000-77	Oceania
000-88	Worldwide

The following table shows more specific mobile operator ID values for carrier-unlocked phones that will be sold in only one country/region. If multiple codes apply, use the regional IDs listed in the preceding table instead. However, if the specific value is used, updates can be targeted more carefully if needed.

MOBILE OPERATOR ID	COUNTRY/REGION
<b>000-AD</b>	Andorra
<b>000-AE</b>	United Arab Emirates
<b>000-AF</b>	Afghanistan
<b>000-AG</b>	Antigua and Barbuda
<b>000-AI</b>	Anguilla
<b>000-AL</b>	Albania
<b>000-AM</b>	Armenia
<b>000-AO</b>	Angola
<b>000-AQ</b>	Antarctica
<b>000-AR</b>	Argentina
<b>000-AS</b>	American Samoa
<b>000-AT</b>	Austria

MOBILE OPERATOR ID	COUNTRY/REGION
000-AU	Australia
000-AW	Aruba
000-AX	Åland Islands
000-AZ	Azerbaijan
000-BA	Bosnia and Herzegovina
000-BB	Barbados
000-BD	Bangladesh
000-BE	Belgium
000-BF	Burkina Faso
000-BG	Bulgaria
000-BH	Bahrain
000-BI	Burundi
000-BJ	Benin
000-BL	Saint Barthélemy
000-BM	Bermuda
000-BN	Brunei
000-BO	Bolivia
000-BQ	Bonaire
000-BR	Brazil

MOBILE OPERATOR ID	COUNTRY/REGION
000-BS	Bahamas, The
000-BT	Bhutan
000-BV	Bouvet Island
000-BW	Botswana
000-BY	Belarus
000-BZ	Belize
000-CA	Canada
000-CC	Cocos (Keeling) Islands
000-CD	Congo (DRC)
000-CF	Central African Republic
000-CG	Congo
000-CH	Switzerland
000-CI	Côte d'Ivoire
000-CK	Cook Islands
000-CL	Chile
000-CM	Cameroon
000-CN	China
000-CO	Colombia
000-CR	Costa Rica

MOBILE OPERATOR ID	COUNTRY/REGION
000-CV	Cabo Verde
000-CW	Curaçao
000-CX	Christmas Island
000-CY	Cyprus
000-CZ	Czech Republic
000-DE	Germany
000-DJ	Djibouti
000-DK	Denmark
000-DM	Dominica
000-DO	Dominican Republic
000-DZ	Algeria
000-EC	Ecuador
000-EE	Estonia
000-EG	Egypt
000-ER	Eritrea
000-ES	Spain
000-ET	Ethiopia
000-FI	Finland
000-FJ	Fiji

MOBILE OPERATOR ID	COUNTRY/REGION
<b>000-FK</b>	Falkland Islands (Islas Malvinas)
<b>000-FM</b>	Micronesia
<b>000-FO</b>	Faroe Islands
<b>000-FR</b>	France
<b>000-GA</b>	Gabon
<b>000-GB</b>	United Kingdom
<b>000-GD</b>	Grenada
<b>000-GE</b>	Georgia
<b>000-GF</b>	French Guiana
<b>000-GG</b>	Guernsey
<b>000-GH</b>	Ghana
<b>000-GI</b>	Gibraltar
<b>000-GL</b>	Greenland
<b>000-GM</b>	Gambia, The
<b>000-GN</b>	Guinea
<b>000-GP</b>	Guadeloupe
<b>000-GQ</b>	Equatorial Guinea
<b>000-GR</b>	Greece
<b>000-GS</b>	South Georgia and the South Sandwich Islands

MOBILE OPERATOR ID	COUNTRY/REGION
<b>000-GT</b>	Guatemala
<b>000-GU</b>	Guam
<b>000-GW</b>	Guinea-Bissau
<b>000-GY</b>	Guyana
<b>000-HK</b>	Hong Kong SAR
<b>000-HM</b>	Heard Island and McDonald Islands
<b>000-HN</b>	Honduras
<b>000-HR</b>	Croatia
<b>000-HT</b>	Haiti
<b>000-HU</b>	Hungary
<b>000-ID</b>	Indonesia
<b>000-IE</b>	Ireland
<b>000-IL</b>	Israel
<b>000-IM</b>	Isle of Man
<b>000-IN</b>	India
<b>000-IO</b>	British Indian Ocean Territory
<b>000-IQ</b>	Iraq
<b>000-IS</b>	Iceland
<b>000-IT</b>	Italy

MOBILE OPERATOR ID	COUNTRY/REGION
<b>000-JE</b>	Jersey
<b>000-JM</b>	Jamaica
<b>000-JO</b>	Jordan
<b>000-JP</b>	Japan
<b>000-KE</b>	Kenya
<b>000-KG</b>	Kyrgyzstan
<b>000-KH</b>	Cambodia
<b>000-KI</b>	Kiribati
<b>000-KM</b>	Comoros
<b>000-KN</b>	Saint Kitts and Nevis
<b>000-KR</b>	Korea
<b>000-KW</b>	Kuwait
<b>000-KY</b>	Cayman Islands
<b>000-KZ</b>	Kazakhstan
<b>000-LA</b>	Laos
<b>000-LB</b>	Lebanon
<b>000-LC</b>	Saint Lucia
<b>000-LI</b>	Liechtenstein
<b>000-LK</b>	Sri Lanka

MOBILE OPERATOR ID	COUNTRY/REGION
000-LR	Liberia
000-LS	Lesotho
000-LT	Lithuania
000-LU	Luxembourg
000-LV	Latvia
000-LY	Libya
000-MA	Morocco
000-MC	Monaco
000-MD	Moldova
000-ME	Montenegro
000-MF	Saint Martin
000-MG	Madagascar
000-MH	Marshall Islands
000-MK	Macedonia, FYRO
000-ML	Mali
000-MM	Myanmar
000-MN	Mongolia
000-MO	Macao SAR
000-MP	Northern Mariana Islands

MOBILE OPERATOR ID	COUNTRY/REGION
000-MQ	Martinique
000-MR	Mauritania
000-MS	Montserrat
000-MT	Malta
000-MU	Mauritius
000-MV	Maldives
000-MW	Malawi
000-MX	Mexico
000-MY	Malaysia
000-MZ	Mozambique
000-NA	Namibia
000-NC	New Caledonia
000-NE	Niger
000-NF	Norfolk Island
000-NG	Nigeria
000-NI	Nicaragua
000-NL	Netherlands
000-NO	Norway
000-NP	Nepal

MOBILE OPERATOR ID	COUNTRY/REGION
<b>000-NR</b>	Nauru
<b>000-NU</b>	Niue
<b>000-NZ</b>	New Zealand
<b>000-OM</b>	Oman
<b>000-PA</b>	Panama
<b>000-PE</b>	Peru
<b>000-PF</b>	French Polynesia
<b>000-PG</b>	Papua New Guinea
<b>000-PH</b>	Philippines
<b>000-PK</b>	Pakistan
<b>000-PL</b>	Poland
<b>000-PM</b>	Saint Pierre and Miquelon
<b>000-PN</b>	Pitcairn Islands
<b>000-PR</b>	Puerto Rico
<b>000-PS</b>	Palestinian Authority
<b>000-PT</b>	Portugal
<b>000-PW</b>	Palau
<b>000-PY</b>	Paraguay
<b>000-QA</b>	Qatar

MOBILE OPERATOR ID	COUNTRY/REGION
<b>000-RE</b>	Reunion
<b>000-RO</b>	Romania
<b>000-RS</b>	Serbia
<b>000-RU</b>	Russia
<b>000-RW</b>	Rwanda
<b>000-SA</b>	Saudi Arabia
<b>000-SB</b>	Solomon Islands
<b>000-SC</b>	Seychelles
<b>000-SE</b>	Sweden
<b>000-SG</b>	Singapore
<b>000-SH</b>	Saint Helena, Ascension, and Tristan da Cunha
<b>000-SI</b>	Slovenia
<b>000-SJ</b>	Svalbard
<b>000-SK</b>	Slovakia
<b>000-SL</b>	Sierra Leone
<b>000-SM</b>	San Marino
<b>000-SN</b>	Senegal
<b>000-SO</b>	Somalia
<b>000-SR</b>	Suriname

MOBILE OPERATOR ID	COUNTRY/REGION
<b>000-SS</b>	South Sudan
<b>000-ST</b>	São Tomé and Príncipe
<b>000-SV</b>	El Salvador
<b>000-SX</b>	Sint Maarten
<b>000-SZ</b>	Swaziland
<b>000-TC</b>	Turks and Caicos Islands
<b>000-TD</b>	Chad
<b>000-TF</b>	French Southern and Antarctic Lands
<b>000-TG</b>	Togo
<b>000-TH</b>	Thailand
<b>000-TJ</b>	Tajikistan
<b>000-TK</b>	Tokelau
<b>000-TL</b>	Timor-Leste
<b>000-TM</b>	Turkmenistan
<b>000-TN</b>	Tunisia
<b>000-TO</b>	Tonga
<b>000-TR</b>	Turkey
<b>000-TT</b>	Trinidad and Tobago
<b>000-TV</b>	Tuvalu

MOBILE OPERATOR ID	COUNTRY/REGION
<b>000-TW</b>	Taiwan
<b>000-TZ</b>	Tanzania
<b>000-UA</b>	Ukraine
<b>000-UG</b>	Uganda
<b>000-UM</b>	US Minor Outlying Islands
<b>000-US</b>	United States
<b>000-UY</b>	Uruguay
<b>000-UZ</b>	Uzbekistan
<b>000-VA</b>	Holy See (Vatican City)
<b>000-VC</b>	Saint Vincent and the Grenadines
<b>000-VE</b>	Venezuela
<b>000-VG</b>	British Virgin Islands
<b>000-VI</b>	US Virgin Islands
<b>000-VN</b>	Vietnam
<b>000-VU</b>	Vanuatu
<b>000-WF</b>	Wallis and Futuna
<b>000-WS</b>	Samoa
<b>000-XE</b>	Sint Eustatius
<b>000-XJ</b>	Jan Mayen

MOBILE OPERATOR ID	COUNTRY/REGION
000-XS	Saba
000-YE	Yemen
000-YT	Mayotte
000-ZA	South Africa
000-ZM	Zambia
000-ZW	Zimbabwe

## Related topics

[Prepare for Windows mobile development](#)

[Customization answer file overview](#)

# Configure power settings

10/2/2018 • 2 minutes to read • [Edit Online](#)

This section contains information about the power settings that you can configure using the Windows provisioning framework. Each power setting topic includes the identification GUID, allowed values, meaning, and common usage scenarios for the setting.

## TIP

The primary audience for these topics is Original Equipment Manufacturers (OEMs). If you're a Windows device owner (consumer) and would like to learn more about power settings in Windows 10, please see [How to enable Hibernate and Sleep in Power Options](#) on Microsoft's community support site. You can also search for troubleshooting instructions on this site if needed.

## Use Windows Configuration Designer to configure power settings

To configure the power settings, you will first create a provisioning package using [Windows Configuration Designer](#). You will then edit the customizations.xml file contained in the package to include your power settings. Use the XML file as one of the inputs to the Windows Configuration Designer command-line to generate either a provisioning package or a Windows image that contains the power settings. For information on how to use the Windows Configuration Designer CLI, see [Use the Windows Configuration Designer command-line interface](#).

The power settings are not visible in the Windows Configuration Designer UI but appear under the main `Common\Power` namespace. This namespace is further divided into various groups including:

- `Policy\Settings` which includes the following subgroups:
  - `AdaptivePowerBehavior`
  - `Processor`
  - `Battery`
  - `Button`
  - `Display`
  - `Disk`
  - `EnergySaver`
  - `PCIExpress`
  - `Sleep`
  - `Misc`
- `Controls\Settings` which includes the following settings:
  - `LidNotificationsAreReliable`

The following example shows what your Windows provisioning answer file might look like after you've written it.

```

<?xml version="1.0" encoding="utf-8"?>
<WindowsCustomizations>
  <PackageConfig xmlns="urn:schemas-Microsoft-com:Windows-ICD-Package-Config.v1.0">
    <ID>{7e5c6cb3-bd16-4c1a-aacb-98c9151d5f20}</ID>  <!-- ID needs to be unique GUID for the package -->
    <Name>CustomOEM.Power.Settings.Control</Name>
    <Version>1.0</Version>
    <OwnerType>OEM</OwnerType>
  </PackageConfig>

  <Settings xmlns="urn:schemas-microsoft-com:windows-provisioning">
    <Customizations>
      <Common>
        <Power>
          <Policy>
            <Settings>
              <Sleep>
                <SchemePersonality>
                  <Default SchemeAlias="Balanced">
                    <Setting>
                      <!-- Duration of time after sleep that the system automatically wakes and
                           enters hibernate in seconds -->
                      <HibernateTimeout>
                        <AcValue>1800</AcValue> <!-- 30 minutes -->
                        <DcValue>1800</DcValue> <!-- 30 minutes -->
                      </HibernateTimeout>
                    </Setting>
                  </Default>
                </SchemePersonality>
              </Sleep>
              <Misc>
                <SchemePersonality>
                  <Default SchemeAlias="Balanced">
                    <Setting>
                      <!-- Enables/Disables only WiFi connection during standby -->
                      <AllowWifiInStandby>
                        <AcValue>0</AcValue>
                        <DcValue>0</DcValue>
                      </AllowWifiInStandby>
                    </Setting>
                  </Default>
                </SchemePersonality>
              </Misc>
            </Settings>
          </Policy>
        </Power>
      </Common>
    </Customizations>
  </Settings>
</WindowsCustomizations>

```

## Use Powercfg.exe to control power schemes

You can use the powercfg.exe tool to control power schemes by providing the GUID or alias for the setting. For more information on how to use this tool, see [Powercfg command-line options](#).

## In this section

TOPIC	DESCRIPTION
-------	-------------

Topic	Description
Adaptive hibernate	Adaptive hibernate supports triggers which eliminate resume to a dead battery, and provide a great Modern Standby experience by ensuring that the system remains in CS for as long as possible.
Power controls	Settings in this subgroup include settings that control the system's power and behavior.
Processor power management options	The Windows 10 processor power management (PPM) algorithms implement OS-level functionality that allows the OS to efficiently use the available processing resources on a platform by balancing the user's expectations of performance and energy efficiency.
Battery settings	Settings in this subgroup control the customization of battery actions and thresholds.
Power button and lid settings	Settings in this subgroup control the customization of system button actions.
Display settings	Settings in this subgroup control the power management of the display.
Disk settings	Settings in this subgroup control the power management of disk devices.
Energy Saver settings	Settings in this subgroup control the battery threshold and brightness when Energy Saver is turned on.
PCI Express settings	Settings in this subgroup control the power management of PCI Express links.
Sleep settings	Settings in this subgroup control sleep, resume, and other related functionality.
Other power settings	Settings in this subgroup do not belong to any other subgroup.
Legacy configuration options	

# Adaptive hibernate

10/9/2018 • 3 minutes to read • [Edit Online](#)

Users can set the Hibernate option in their Windows devices to put the system into a low power state when the system is not in use. The current logic for hibernate relies on an OEM- or user-configured doze to hibernate timer. The most common timer value is 4 hours. A fixed doze to hibernate timer may offer a consistent and predictable user experience, however it doesn't address rapid drain of battery.

The timer-based logic has some significant user experience drawbacks. A fixed doze timer can result in the system fully draining the battery in standby if it happened within the doze timeout or cut short a Modern Standby experience by hibernating at doze timeout. The timer is generally not the best option when it comes to addressing the worst case battery drain and the system needs to be adaptive and hibernate based on battery drain and user needs.

Adaptive hibernate provides triggers which allow the system to hibernate intelligently. These triggers provide the following benefits:

- Eliminate resuming to a dead battery.
- Provide a great [Modern Standby](#) (MS) experience by ensuring that the system remains in MS for as long as possible.

To support the adaptive hibernate triggers, the system is enabled with default values. However, OEMs can program these triggers to ensure that machines hibernate to provide the best possible experience to users.

## System requirements

The triggers apply to Modern Standby systems only.

## Default behavior

Machines will have adaptive hibernate timeout enabled by default; however, OEMs can configure the settings using a provisioning package file. See the following sections for more information on how to do this.

## Hibernate triggers

Adaptive hibernate settings (standby budget setting and standby reserve time setting) are exposed as hidden power settings. The settings are applied on DC only and have no impact on AC.

### Standby budget setting

The following table lists the settings you can use to set the standby budget, which is the amount of battery the user is allowed to drain during standby.

BUDGET SETTING	DEFINITION	EXPOSED AS	POWERCFG COMMAND

BUDGET SETTING	DEFINITION	EXPOSED AS	POWERCFG COMMAND
StandbyBudgetPercent	Defines the battery drain percentage that the user is allowed in a 24-hour standby period. If the drain percentage is reached, the device transitions to Hibernate. Default is 5%.	Power setting	<pre>powercfg /setdcvalueindex scheme_current sub_presence standbybudgetpercent</pre>

You can also configure these settings using a custom provisioning package file for OEM images. For more information about powercfg, see [Powercfg command-line options](#).

### Standby reserve time setting

Reserve time is the amount of time the user is guaranteed to have the screen on after the system resumes from standby or hibernate. The following table lists the settings you can use to set the reserve time.

BUDGET SETTING	DEFINITION	EXPOSED AS	POWERCFG COMMAND
StandbyReserveTime	Defines the screen on time, in seconds, that will be available to the user after standby exits and the screen turns on. Default is 1200 seconds.	Power setting	<pre>powercfg /setdcvalueindex scheme_current sub_presence standbyreservetime</pre>

You can also configure these settings using a custom provisioning package file for OEM images. For more information about powercfg, see [Powercfg command-line options](#).

## Windows provisioning package sample

You can use the Windows Provisioning framework to configure the adaptive hibernate settings described in this section. First, create a provisioning package using [Windows Configuration Designer](#). You will then edit the customizations.xml file contained in the package to include your power settings, which appear under the `Common\Power\Policy\Settings\AdaptivePowerBehavior` namespace. Use the XML file as one of the inputs to the Windows Configuration Designer command-line interface to generate either a provisioning package that contains the power settings. You can then apply the provisioning package to the image. For information on how to use the Windows Configuration Designer CLI, see [Use the Windows Configuration Designer command-line interface](#).

The following example shows what your Windows provisioning answer file might look like after you've written it to configure adaptive hibernate settings.

```

<?xml version="1.0" encoding="utf-8"?>
<WindowsCustomizations>
    <PackageConfig xmlns="urn:schemas-Microsoft-com:Windows-ICD-Package-Config.v1.0">
        <ID>{XXXX GUID}</ID> <!-- ID needs to be unique GUID for the package -->
        <Name>CustomOEM.Power.Settings.Control</Name>
        <Version>1.0</Version>
        <OwnerType>OEM</OwnerType>
    </PackageConfig>

    <Settings xmlns="urn:schemas-microsoft-com:windows-provisioning">
        <Customizations>
            <Common>
                <Power>
                    <Policy>
                        <Settings>
                            <AdaptivePowerBehavior>
                                <SchemePersonality>
                                    <Default SchemeAlias="Balanced">
                                        <Setting>
                                            <!-- After entering standby, battery drain percentage allowed before the device transitions to hibernate -->
                                            <StandbyBudgetPercent>
                                                <DcValue>3</DcValue>
                                            </StandbyBudgetPercent>
                                            <!-- After entering standby, number of seconds before the device automatically transitions to hibernate -->
                                            <StandbyReserveTime>
                                                <DcValue>600</DcValue>
                                            </StandbyReserveTime>
                                        </Setting>
                                    </Default>
                                </SchemePersonality>
                            </AdaptivePowerBehavior>
                        </Settings>
                    </Policy>
                </Power>
            </Common>
        </Customizations>
    </Settings>

```

## User prediction

User usage prediction no longer triggers Hibernate. This is a change from previous versions of Windows. Windows continues to support automatically transitioning from Hibernate back to Modern Standby based on user prediction, however this requires that the device implement RTCWake or the Time & Device Alarm object in ACPI.

# StandbyBudgetPercent

10/2/2018 • 2 minutes to read • [Edit Online](#)

Defines the battery drain percentage that the user is allowed in a standby session.

## Aliases and setting visibility

- **Windows Provisioning:** StandbyBudgetPercent
- **Hidden setting:** Yes

## Values

The value denotes the percentage, example: 3 = 3%.

You can configure the values for the following sub-settings: DcValue and AcValue

## Applies to

Available in Windows 10, version 1607 and later versions of Windows.

# StandbyReserveTime

10/2/2018 • 2 minutes to read • [Edit Online](#)

Defines the screen on time, in seconds, that will be available to the user after standby exists and the screen turns on.

## Aliases and setting visibility

- **Windows Provisioning:** StandbyReserveTime
- **Hidden setting:** Yes

## Values

The value denotes the time, in seconds.

You can configure the values for the following sub-settings: DcValue and AcValue

## Applies to

Available in Windows 10, version 1607 and later versions of Windows.

# Power controls

10/2/2018 • 2 minutes to read • [Edit Online](#)

Settings in this subgroup include settings that control the system's power and behavior.

## Subgroup, path, and setting visibility

- **Subgroup:** Controls settings
- **Windows provisioning path:** `Common\Power\Controls\Settings`
- **Hidden setting:** Yes

## In this section

TOPIC	DESCRIPTION
<a href="#">EnableInputSuppression</a>	New in Windows 10, version 1803. Use to enable input suppression on a Modern Standby system with a clamshell form factor when the lid is closed, there is no external monitor connected, and the system is on DC power.
<a href="#">IgnoreCsComplianceCheck</a>	New in Windows 10, version 1803. Use to disable the default OS requirement of having non-rotational media in a Modern Standby system.
<a href="#">LidNotificationsAreReliable</a>	Use to notify the OS whether the platform guarantees that lid notifications are sent whenever the lid is opened or closed.

# EnableInputSuppression

10/2/2018 • 2 minutes to read • [Edit Online](#)

Use to enable input suppression on a Modern Standby system with a clamshell form factor when the lid is closed, there is no external monitor connected, and the system is on DC power.

When the conditions above are met, it is expected that the system will stay in a low power state to preserve battery life. However, some input devices can wake the system from standby even if the user is not using them. For example, a Bluetooth mouse paired with the system may be stored inside a laptop bag with the system, and the motion of the mouse causes the system to wake. Enabling input suppression prevents this behavior.

## Aliases and setting visibility

- **Windows Provisioning:** `EnableInputSuppression`
- **Hidden setting:** Yes

## Values

VALUE	DESCRIPTION
1	Enable input suppression.
0	Disable input suppression (default).

## Applies to

Available in Windows 10, version 1803 and later versions of Windows.

# IgnoreCsComplianceCheck

10/2/2018 • 2 minutes to read • [Edit Online](#)

Use to disable the default OS requirement of having non-rotational media in a Modern Standby system.

## Aliases and setting visibility

- **Windows Provisioning:** `IgnoreCsComplianceCheck`
- **Hidden setting:** Yes

## Values

VALUE	DESCRIPTION
1	Disable the check requiring non-rotational media in a Modern Standby system.
0	Enable the check requiring non-rotational media in a Modern Standby system (default).

## Remarks

Enabling Modern Standby on a system with rotational storage media is not recommended, as this may result in increased power consumption due to the tradeoff between power cycling and hard drive reliability. It may also result in higher exit latency upon resume from Modern Standby (compared to the latency with SSDs). Please refer to the [Modern Standby rotational storage guidelines](#) for more information.

## Applies to

Available in Windows 10, version 1803 and later versions of Windows.

# LidNotificationsAreReliable

10/2/2018 • 2 minutes to read • [Edit Online](#)

Use to notify the OS whether the platform guarantees that lid notifications are sent whenever the lid is opened or closed.

## Aliases and setting visibility

- **Windows Provisioning:** `LidNotificationsAreReliable`
- **Hidden setting:** Yes

## Values

VALUE	DESCRIPTION
True	The platform guarantees that lid notifications will be sent every time the device lid is opened or closed. The OS suppresses Windows Hello when the device lid is closed to ensure further input is not processed and to save battery life. OEMs must reliably report lid open and lid close events to opt-in to this setting. If there are scenarios where a lid open event is not reliably reported to the OS, Windows Hello may not work for the user.
False	The platform does not guarantee that lid notifications are sent every time the device lid is opened or closed.

## Remarks

Depending on your platform scenarios, you may also want to set the `LidOpenWake` setting ([Lid open wake action](#)).

For example:

- If you want to implement a platform that does nothing when the lid is opened, but you want to suppress Windows Hello when the lid is closed, you'll want to set `LidOpenWake`=0 and `LidNotificationsAreReliable`=True.
- If you have a device that has a rigid keyboard and the risk of the lid opening and causing the device to turn on is low, you may want to implement a platform that turns on the display when the lid is opened, but you want to suppress Windows Hello when the lid is closed, you'll want to set `LidOpenWake`=1 and `LidNotificationsAreReliable`=True.

## Applies to

Available in Windows 10, version 1607 and later versions of Windows.

# Processor power management options

10/2/2018 • 2 minutes to read • [Edit Online](#)

The Windows 10 processor power management (PPM) algorithms implement OS-level functionality that allows the OS to efficiently use the available processing resources on a platform by balancing the user's expectations of performance and energy efficiency.

The algorithms have the following characteristics:

- They scale from big servers to tablet form factors.
- They are customizable through a statically configurable power policy infrastructure.
- They are hierarchical and abstracted in a manner that separates platform-agnostic portions of the algorithms from platform-specific portions.

At a high-level, the Windows PPM is made up of the following parts:

- **Core parking engine** - Makes global scalability decisions about the workload and determines the optimum set of compute cores to execute with.
- **Performance state engine** - Makes per-processor performance scaling decisions.
- **Platform specific controls** - Implements the mechanics of state transitions and optionally provides feedback about the effectiveness of OS state decisions and runtime platform constraints.

IHV partners can enable preliminary validation and measurement of the effects of the policy controls on different hardware configurations.

## Power profiles

You can use the Windows Provisioning framework to configure the processor power settings described in this section. First, create a provisioning package using [Windows Configuration Designer](#). You will then edit the customizations.xml file contained in the package to include your power settings, which appear under the `Common\Power\Policy\Settings\Processor` namespace. Use the XML file as one of the inputs to the Windows Configuration Designer command-line interface to generate either a provisioning package that contains the power settings. You can then apply the provisioning package to the image. For information on how to use the Windows Configuration Designer CLI, see [Use the Windows Configuration Designer command-line interface](#).

The processor namespace is divided into three sets of identical power processor configurations called power profiles. The power profiles are used by the power processor engine to adapt the performance and parking algorithm on various system use cases.

Windows 10 supports the following profiles:

- *Default* profile is the configuration set that is active most of the time.
- *LowLatency* is the profile that is activated during boot and during app launch time.
- *LowPower* is the profile that is activated during the buffering phase of media playback scenarios.
- *Constrained* is a profile activated by the battery saver feature on Windows 10 for desktop editions (Home, Pro, Enterprise, and Education). This is not available on Windows 10 Mobile.

Each profile supports the following configuration settings:

- [CPMinCores](#)
- [CPMaxCores](#)
- [CPIIncreaseTime](#)

- [CPDecreaseTime](#)
- [CPConcurrency](#)
- [CPDistribution](#)
- [CPHeadroom](#)
- [CpLatencyHintUnpark](#)
- [MaxPerformance](#)
- [MinPerformance](#)
- [PerfIncreaseThreshold](#)
- [PerfIncreaseTime](#)
- [PerfDecreaseThreshold](#)
- [PerfDecreaseTime](#)
- [PerfLatencyHint](#)
- [PerfAutonomousMode](#)
- [PerfEnergyPreference](#)

On systems with processors with heterogeneous architecture, the configuration settings for efficiency class 1 cores use a similar naming convention. Efficiency class is defined in ACPI 6.0 section 5.2.12.14 GICC Structure. For more information, consult the ACPI specification.

The common parameters have the suffix "1" to indicate efficiency class. Hetero-specific parameters have the prefix "Hetero".

- [CPMinCores1](#)
- [CPMaxCores1](#)
- [HeteroIncreaseTime](#)
- [HeteroDecreaseTime](#)
- [HeteroIncreaseThreshold](#)
- [HeteroDecreaseThreshold](#)
- [CpLatencyHintUnpark1](#)
- [MaxPerformance1](#)
- [MinPerformance1](#)
- [PerfIncreaseThreshold1](#)
- [PerfIncreaseTime1](#)
- [PerfDecreaseThreshold1](#)
- [PerfDecreaseTime1](#)
- [PerfLatencyHint1](#)
- [HeteroClass1InitialPerf](#)
- [HeteroClass0FloorPerf](#)

# Static configuration options for core parking

10/2/2018 • 2 minutes to read • [Edit Online](#)

You can use the static configuration options documented in this section to tune the behavior of the core parking engine.

## In this section

TOPIC	DESCRIPTION
<a href="#">CPMinCores</a>	<code>CPMinCores</code> specifies the minimum percentage of logical processors (in terms of all logical processors that are enabled on the system within each NUMA node) that can be placed in the un-parked state at any given time.
<a href="#">CPMaxCores</a>	<code>CPMaxCores</code> specifies the maximum percentage of logical processors (in terms of logical processors within each NUMA node) that can be in the un-parked state at any given time.
<a href="#">CPIIncreaseTime</a>	<code>CPIIncreaseTime</code> specifies the minimum amount of time that must elapse before additional logical processors can be transitioned from the parked state to the unparked state. The time is specified in units of the number of processor performance time check intervals.
<a href="#">CPDecreaseTime</a>	<code>CPDecreaseTime</code> specifies the minimum amount of time that must elapse before additional logical processors can be transitioned from the unparked state to the parked state. The time is specified in units of the number of processor performance time check intervals.
<a href="#">CPConcurrency</a>	<code>CPConcurrency</code> specifies the threshold for determining concurrency of the node.
<a href="#">CPDistribution</a>	<code>CPDistribution</code> specifies the utilization, in percentage, to use in the concurrency distribution to select the number of logical processors to distribute utility to.
<a href="#">CPHeadroom</a>	<code>CPHeadroom</code> specifies the value of utilization that would cause the core parking engine to unpark an additional logical processor if the least utilized processor out of the unparked set of processors had more utilization. This enables increases in concurrency to be detected.

TOPIC	DESCRIPTION
<a href="#">CpLatencyHintUnpark</a>	<code>CPLatencyHintUnpark</code> specifies the minimum number of unparked cores when a system low latency hint is detected.

# CPMinCores

10/2/2018 • 2 minutes to read • [Edit Online](#)

`CPMinCores` specifies the minimum percentage of logical processors (in terms of all logical processors that are enabled on the system within each NUMA node) that can be placed in the un-parked state at any given time.

For example, in a NUMA node with 16 logical processors, configuring the value of this setting to 25% ensures that at least 4 logical processors are always in the un-parked state. The Core Parking algorithm is disabled if the value of this setting is 100%.

## Aliases and setting visibility

- **Windows Provisioning:** `CPMinCores`, `CPMinCores1`
- **PowerCfg:** `CPMINCORES`, `CPMINCORES1`
- **Hidden setting:** Yes

## Values

The value denotes percentage (%).

Minimum value	0
Maximum value	100

## Applies to

WINDOWS EDITION	X86-BASED DEVICES	X64-BASED DEVICES	ARM-BASED DEVICES
Windows 10 for desktop editions (Home, Pro, Enterprise, and Education)	x86	amd64	N/A
Windows 10 Mobile	N/A	N/A	Supported

# CPMaxCores

10/2/2018 • 2 minutes to read • [Edit Online](#)

`CPMaxCores` specifies the maximum percentage of logical processors (in terms of logical processors within each NUMA node) that can be in the un-parked state at any given time.

For example, in a NUMA node with 16 logical processors, configuring the value of this setting to 50% ensures that no more than 8 logical processors are ever in the un-parked state at the same time. The value of this setting will automatically be rounded up to the value of `CPMinCores`.

## Aliases and setting visibility

- **Windows Provisioning:** `CPMaxCores`, `CPMaxCores1`
- **PowerCfg:** `CPMAXCORES`, `CPMAXCORES1`
- **Hidden setting:** Yes

## Values

The value denotes percentage (%).

Minimum value	0
Maximum value	100

## Applies to

WINDOWS EDITION	X86-BASED DEVICES	X64-BASED DEVICES	ARM-BASED DEVICES
Windows 10 for desktop editions (Home, Pro, Enterprise, and Education)	x86	amd64	N/A
Windows 10 Mobile	N/A	N/A	Supported

# CPIncreaseTime

10/2/2018 • 2 minutes to read • [Edit Online](#)

`CPIncreaseTime` specifies the minimum amount of time that must elapse before additional logical processors can be transitioned from the parked state to the unparked state. The time is specified in units of the number of processor performance time check intervals.

## Aliases and setting visibility

- **Windows Provisioning:** `CPIncreaseTime`
- **PowerCfg:** `CPINCREASETIME`
- **Hidden setting:** Yes

## Values

The value denotes time check intervals.

Minimum value	0
Maximum value	100

## Applies to

WINDOWS EDITION	X86-BASED DEVICES	X64-BASED DEVICES	ARM-BASED DEVICES
Windows 10 for desktop editions (Home, Pro, Enterprise, and Education)	x86	amd64	N/A
Windows 10 Mobile	N/A	N/A	Supported

# CPDecreaseTime

10/2/2018 • 2 minutes to read • [Edit Online](#)

`CPDecreaseTime` specifies the minimum amount of time that must elapse before additional logical processors can be transitioned from the unparked state to the parked state. The time is specified in units of the number of processor performance time check intervals.

## Aliases and setting visibility

- **Windows Provisioning:** `CPDecreaseTime`
- **PowerCfg:** `CPDECREASETIME`
- **Hidden setting:** Yes

## Values

The value denotes time check intervals.

Minimum value	0
Maximum value	100

## Applies to

WINDOWS EDITION	X86-BASED DEVICES	X64-BASED DEVICES	ARM-BASED DEVICES
Windows 10 for desktop editions (Home, Pro, Enterprise, and Education)	x86	amd64	N/A
Windows 10 Mobile	N/A	N/A	Supported

# CPConcurrency

10/2/2018 • 2 minutes to read • [Edit Online](#)

`CPConcurrency` specifies the threshold for determining concurrency of the node.

## Aliases and setting visibility

- **Windows Provisioning:** `CPConcurrency`
- **PowerCfg:** `CPCONCURRENCY`
- **Hidden setting:** Yes

## Values

The value denotes percentage (%).

Minimum value	0
Maximum value	100

## Applies to

WINDOWS EDITION	X86-BASED DEVICES	X64-BASED DEVICES	ARM-BASED DEVICES
Windows 10 for desktop editions (Home, Pro, Enterprise, and Education)	x86	amd64	N/A
Windows 10 Mobile	N/A	N/A	Supported

# CPDistribution

10/2/2018 • 2 minutes to read • [Edit Online](#)

`CPDistribution` specifies the utilization, in percentage, to use in the concurrency distribution to select the number of logical processors to distribute utility to. This may be fewer, but never greater, than the number of logical processors that are selected to be unparked.

## Aliases and setting visibility

- **Windows Provisioning:** `CPDistribution`
- **PowerCfg:** `CPDISTRIBUTION`
- **Hidden setting:** Yes

## Values

The value denotes percentage (%).

Minimum value	0
Maximum value	100

## Applies to

WINDOWS EDITION	X86-BASED DEVICES	X64-BASED DEVICES	ARM-BASED DEVICES
Windows 10 for desktop editions (Home, Pro, Enterprise, and Education)	x86	amd64	N/A
Windows 10 Mobile	N/A	N/A	Supported

# CPHeadroom

10/2/2018 • 2 minutes to read • [Edit Online](#)

`CPHeadroom` specifies the value of utilization that would cause the core parking engine to unpark an additional logical processor if the least utilized processor out of the unparked set of processors had more utilization. This enables increases in concurrency to be detected.

## Aliases and setting visibility

- **Windows Provisioning:** `CPHeadroom`
- **PowerCfg:** `CPHEADROOM`
- **Hidden setting:** Yes

## Values

The value denotes percentage (%).

Minimum value	0
Maximum value	100

## Applies to

WINDOWS EDITION	X86-BASED DEVICES	X64-BASED DEVICES	ARM-BASED DEVICES
Windows 10 for desktop editions (Home, Pro, Enterprise, and Education)	x86	amd64	N/A
Windows 10 Mobile	N/A	N/A	Supported

# CpLatencyHintUnpark

10/2/2018 • 2 minutes to read • [Edit Online](#)

`CPLatencyHintUnpark` specifies the minimum number of unparked cores when a system low latency hint is detected.

## Aliases and setting visibility

- **Windows Provisioning:** `CpLatencyHintUnpark`, `CpLatencyHintUnpark1`
- **PowerCfg:** `LATENCYHINTUNPARK`, `LATENCYHINTUNPARK1`
- **Hidden setting:** Yes

## Values

The value denotes percentage (%).

Minimum value	0
Maximum value	100

## Applies to

WINDOWS EDITION	X86-BASED DEVICES	X64-BASED DEVICES	ARM-BASED DEVICES
Windows 10 for desktop editions (Home, Pro, Enterprise, and Education)	x86	amd64	N/A
Windows 10 Mobile	N/A	N/A	Supported

# Static configuration options for the performance state engine

10/2/2018 • 2 minutes to read • [Edit Online](#)

You can use the static configuration options documented in this section to tune the behavior of the performance state selection algorithms.

## In this section

TOPIC	DESCRIPTION
<a href="#">MaxPerformance</a>	<code>MaxPerformance</code> specifies the maximum processor performance state, which is specified as a percentage of maximum processor performance.
<a href="#">MinPerformance</a>	<code>MinPerformance</code> specifies the minimum processor performance state, which is specified as a percentage of maximum processor performance.
<a href="#">PerfIncreaseThreshold</a>	<code>PerfIncreaseThreshold</code> specifies the percentage of processor utilization, in terms of the maximum processor utilization, that is required to increase the processor to a higher performance state.
<a href="#">PerfIncreaseTime</a>	<code>PerfIncreaseTime</code> specifies minimum amount of time that must elapse between subsequent increases in the processor performance state. The time is specified in units of the number of processor performance time check intervals.
<a href="#">PerfDecreaseThreshold</a>	<code>PerfDecreaseThreshold</code> specifies the percentage of processor utilization, in terms of the maximum processor utilization, that is required to reduce the processor to a lower performance state.
<a href="#">PerfDecreaseTime</a>	<code>PerfDecreaseTime</code> specifies minimum amount of time that must elapse between subsequent reductions in the processor performance state. The time is specified in units of the number of processor performance time check intervals.

TOPIC	DESCRIPTION
<a href="#">PerfLatencyHint</a>	<p><code>PerfLatencyHint</code> specifies the processor performance in response to latency sensitivity hints. Such hints are generated when an event preceding an expected latency-sensitive operation is detected. Examples include mouse button up events (for all mouse buttons), touch gesture start and gesture stop (finger down and finger up), and keyboard enter key down.</p>
<a href="#">PerfAutonomousMode</a>	<p><code>PerfAutonomousMode</code> controls whether autonomous mode is enabled on systems that implement version 2 of the CPPC interface, and determines whether desired performance requests should be provided to the platform. On systems with other performance state interfaces, this setting has no effect.</p>
<a href="#">PerfEnergyPreference</a>	<p><code>PerfEnergyPreference</code> specifies the value to program in the energy performance preference register on systems that implement version 2 of the CPPC interface.</p>
<a href="#">PerfAutonomousWindow</a>	<p><code>PerfAutonomousWindow</code> specifies the value to program in the autonomous activity window register on systems that implement version 2 of the CPPC interface and have autonomous mode enabled. Longer values indicate to the platform that it should be less sensitive to short duration spikes/dips in processor utilization.</p>
<a href="#">DutyCycling</a>	<p><code>DutyCycling</code> enables or disables the duty cycling capability on systems that support processor duty cycling.</p>

# MaxPerformance

10/2/2018 • 2 minutes to read • [Edit Online](#)

`MaxPerformance` specifies the maximum processor performance state, which is specified as a percentage of maximum processor performance.

## Aliases and setting visibility

- **Windows Provisioning:** `MaxPerformance`, `MaxPerformance1`
- **PowerCfg:** `PROCTHROTTLEMAX`, `PROCTHROTTLEMAX1`
- **Hidden setting:** No

## Values

The value denotes percentage (%).

Minimum value	0
Maximum value	100

## Applies to

WINDOWS EDITION	X86-BASED DEVICES	X64-BASED DEVICES	ARM-BASED DEVICES
Windows 10 for desktop editions (Home, Pro, Enterprise, and Education)	x86	amd64	N/A
Windows 10 Mobile	N/A	N/A	Supported

# MinPerformance

10/2/2018 • 2 minutes to read • [Edit Online](#)

`MinPerformance` specifies the minimum processor performance state, which is specified as a percentage of maximum processor performance.

## Aliases and setting visibility

- **Windows Provisioning:** `MinPerformance`, `MinPerformance1`
- **PowerCfg:** `PROCTHROTTLEMIN`, `PROCTHROTTLEMIN1`
- **Hidden setting:** No

## Values

The value denotes percentage (%).

Minimum value	0
Maximum value	100

## Applies to

WINDOWS EDITION	X86-BASED DEVICES	X64-BASED DEVICES	ARM-BASED DEVICES
Windows 10 for desktop editions (Home, Pro, Enterprise, and Education)	x86	amd64	N/A
Windows 10 Mobile	N/A	N/A	Supported

# PerfIncreaseThreshold

10/2/2018 • 2 minutes to read • [Edit Online](#)

`PerfIncreaseThreshold` specifies the percentage of processor utilization, in terms of the maximum processor utilization, that is required to increase the processor to a higher performance state.

## Aliases and setting visibility

- **Windows Provisioning:** `PerfIncreaseThreshold`, `PerfIncreaseThreshold1`
- **PowerCfg:** `PERFINCThreshold`, `PERFINCThreshold1`
- **Hidden setting:** Yes

## Values

The value denotes percentage (%).

Minimum value	0
Maximum value	100

## Applies to

WINDOWS EDITION	X86-BASED DEVICES	X64-BASED DEVICES	ARM-BASED DEVICES
Windows 10 for desktop editions (Home, Pro, Enterprise, and Education)	x86	amd64	N/A
Windows 10 Mobile	N/A	N/A	Supported

# PerfIncreaseTime

10/2/2018 • 2 minutes to read • [Edit Online](#)

`PerfIncreaseTime` specifies minimum amount of time that must elapse between subsequent increases in the processor performance state. The time is specified in units of the number of processor performance time check intervals.

## Aliases and setting visibility

- **Windows Provisioning:** `PerfIncreaseTime`, `PerfIncreaseTime1`
- **PowerCfg:** `PERFINCTIME`, `PERFINCTIME1`
- **Hidden setting:** Yes

## Values

The value denotes time check intervals.

Minimum value	0
Maximum value	100

## Applies to

WINDOWS EDITION	X86-BASED DEVICES	X64-BASED DEVICES	ARM-BASED DEVICES
Windows 10 for desktop editions (Home, Pro, Enterprise, and Education)	x86	amd64	N/A
Windows 10 Mobile	N/A	N/A	Supported

# PerfDecreaseThreshold

10/2/2018 • 2 minutes to read • [Edit Online](#)

`PerfDecreaseThreshold` specifies the percentage of processor utilization, in terms of the maximum processor utilization, that is required to reduce the processor to a lower performance state.

## Aliases and setting visibility

- **Windows Provisioning:** `PerfDecreaseThreshold`, `PerfDecreaseThreshold1`
- **PowerCfg:** `PERFDECTHRESHOLD`, `PERFDECTHRESHOLD1`
- **Hidden setting:** Yes

## Values

The value denotes percentage (%).

Minimum value	0
Maximum value	100

## Applies to

WINDOWS EDITION	X86-BASED DEVICES	X64-BASED DEVICES	ARM-BASED DEVICES
Windows 10 for desktop editions (Home, Pro, Enterprise, and Education)	x86	amd64	N/A
Windows 10 Mobile	N/A	N/A	Supported

# PerfDecreaseTime

10/2/2018 • 2 minutes to read • [Edit Online](#)

`PerfDecreaseTime` specifies minimum amount of time that must elapse between subsequent reductions in the processor performance state. The time is specified in units of the number of processor performance time check intervals.

## Aliases and setting visibility

- **Windows Provisioning:** `PerfDecreaseTime`, `PerfDecreaseTime1`
- **PowerCfg:** `PERFDECTIME`, `PERFDECTIME1`
- **Hidden setting:** Yes

## Values

The value denotes time check intervals.

Minimum value	0
Maximum value	100

## Applies to

WINDOWS EDITION	X86-BASED DEVICES	X64-BASED DEVICES	ARM-BASED DEVICES
Windows 10 for desktop editions (Home, Pro, Enterprise, and Education)	x86	amd64	N/A
Windows 10 Mobile	N/A	N/A	Supported

# PerfLatencyHint

10/2/2018 • 2 minutes to read • [Edit Online](#)

`PerfLatencyHint` specifies the processor performance in response to latency sensitivity hints. Such hints are generated when an event preceding an expected latency-sensitive operation is detected. Examples include mouse button up events (for all mouse buttons), touch gesture start and gesture stop (finger down and finger up), and keyboard enter key down.

When set to 0, the processor performance engine does not take latency sensitivity hints into account when selecting a performance state. Otherwise, the performance is raised system-wide to the specified performance level.

## Aliases and setting visibility

- **Windows Provisioning:** `PerfLatencyHint`, `PerfLatencyHint1`
- **PowerCfg:** `LATENCYHINTPERF`, `LATENCYHINTPERF1`
- **Hidden setting:** Yes

## Values

The value denotes percentage (%).

Minimum value	0
Maximum value	100

## Applies to

WINDOWS EDITION	X86-BASED DEVICES	X64-BASED DEVICES	ARM-BASED DEVICES
Windows 10 for desktop editions (Home, Pro, Enterprise, and Education)	x86	amd64	N/A
Windows 10 Mobile	N/A	N/A	Supported

# PerfAutonomousMode

10/2/2018 • 2 minutes to read • [Edit Online](#)

`PerfAutonomousMode` controls whether autonomous mode is enabled on systems that implement version 2 of the CPPC interface, and determines whether desired performance requests should be provided to the platform. On systems with other performance state interfaces, this setting has no effect.

## Note

Platforms that support CPPC version 2 may only support autonomous disabled or autonomous enabled mode. If only one mode is supported, the OS uses that mode and ignores the `PerfAutonomousMode` power setting.

## Aliases and setting visibility

- **Windows Provisioning:** `PerfAutonomousMode`
- **PowerCfg:** `PERFAUTONOMOUS`
- **Hidden setting:** Yes

## Values

INDEX	NAME	DESCRIPTION
0	Disabled	The performance state engine disables autonomous mode, determines desired performance levels, and conveys those performance levels to the platform.
1	Enabled	The performance state engine enables autonomous mode and stops providing desired performance levels to the platform.

## Applies to

WINDOWS EDITION	X86-BASED DEVICES	X64-BASED DEVICES	ARM-BASED DEVICES
Windows 10 for desktop editions (Home, Pro, Enterprise, and Education)	x86	amd64	N/A
Windows 10 Mobile	N/A	N/A	Supported

# PerfEnergyPreference

10/2/2018 • 2 minutes to read • [Edit Online](#)

`PerfEnergyPreference` specifies the value to program in the energy performance preference register on systems that implement version 2 of the CPPC interface.

When set to 0, the energy performance preference register is programmed to 0 to favor performance. When set to 100, the energy performance preference register is set to 255 to favor energy savings. When set to an intermediate value, the energy performance preference register is programmed to the value: (setting \* 255) / 100.

## Aliases and setting visibility

- **Windows Provisioning:** `PerfEnergyPreference`
- **PowerCfg:** `PERFEPP`
- **Hidden setting:** Yes

## Values

The value denotes percentage (%).

Minimum value	0
Maximum value	100

## Applies to

WINDOWS EDITION	X86-BASED DEVICES	X64-BASED DEVICES	ARM-BASED DEVICES
Windows 10 for desktop editions (Home, Pro, Enterprise, and Education)	x86	amd64	N/A
Windows 10 Mobile	N/A	N/A	Supported

# PerfAutonomousWindow

10/2/2018 • 2 minutes to read • [Edit Online](#)

`PerfAutonomousWindow` specifies the value to program in the autonomous activity window register on systems that implement version 2 of the CPPC interface and have autonomous mode enabled. Longer values indicate to the platform that it should be less sensitive to short duration spikes/dips in processor utilization.

## Aliases and setting visibility

- **Windows Provisioning:** `PerfAutonomousWindow`
- **PowerCfg:** `PERFAUTONOMOUSWINDOW`
- **Hidden setting:** Yes

## Values

The value denotes microseconds.

Minimum value	0
Maximum value	1,270,000,000

## Applies to

WINDOWS EDITION	X86-BASED DEVICES	X64-BASED DEVICES	ARM-BASED DEVICES
Windows 10 for desktop editions (Home, Pro, Enterprise, and Education)	x86	amd64	N/A
Windows 10 Mobile	N/A	N/A	Supported

# DutyCycling

10/2/2018 • 2 minutes to read • [Edit Online](#)

`DutyCycling` enables or disables the duty cycling capability on systems that support processor duty cycling.

## Aliases and setting visibility

- **Windows Provisioning:** `DutyCycling`
- **PowerCfg:** `PERFDUTYCYLING`
- **Hidden setting:** Yes

## Values

INDEX	NAME	DESCRIPTION
0	Disabled	Processor duty cycling is not allowed.
1	Enabled	Processor duty cycling is allowed.

## Applies to

WINDOWS EDITION	X86-BASED DEVICES	X64-BASED DEVICES	ARM-BASED DEVICES
Windows 10 for desktop editions (Home, Pro, Enterprise, and Education)	x86	amd64	N/A
Windows 10 Mobile	N/A	N/A	Supported

# Static configuration options for heterogeneous power scheduling

10/2/2018 • 2 minutes to read • [Edit Online](#)

You can use the static configuration options documented in this section to tune the core parking engine on heterogeneous systems.

**Note** These settings are only valid for class 1 cores and replace CP\_CONCURRENCY, PARK\_DISTRIBUTION\_THRESHOLD and CP\_HEADROOM.

## In this section

TOPIC	DESCRIPTION
<a href="#">HeteroIncreaseThreshold</a>	<code>HeteroIncreaseThreshold</code> specifies the threshold value to cross above, which is required to unpark the Nth efficiency class 1 core. There is a separate value for each core index. The threshold is relative to efficiency class 0 performance.
<a href="#">HeteroDecreaseThreshold</a>	<code>HeteroDecreaseThreshold</code> specifies a threshold to cross below, which is required to park the Nth efficiency class 1 core. There is a separate value for each core index. The threshold is relative to efficiency class 0 performance.
<a href="#">HeteroIncreaseTime</a>	<code>HeteroIncreaseTime</code> specifies the minimum amount of time that must elapse before additional efficiency class 1 logical processors can be transitioned from the parked state to the unparked state. The time is specified in processor performance time check intervals.
<a href="#">HeteroDecreaseTime</a>	<code>HeteroDecreaseTime</code> specifies the minimum amount of time that must elapse before additional efficiency class 1 logical processors can be transitioned from the unparked state to the parked state. The time is specified in performance time check intervals.
<a href="#">HeteroClass1InitialPerf</a>	<code>HeteroClass1InitialPerf</code> specifies the initial performance percentage of the efficiency class 1 core when this core is unparked.
<a href="#">HeteroClass0FloorPerf</a>	<code>HeteroClass0FloorPerf</code> specifies the performance level floor, in percentage, to use for efficiency class 0 processors if there is at least one unparked efficiency class 1 processor.

# HeteroIncreaseThreshold

10/2/2018 • 2 minutes to read • [Edit Online](#)

`HeteroIncreaseThreshold` specifies the threshold value to cross above, which is required to unpark the Nth efficiency class 1 core. There is a separate value for each core index. The threshold is relative to efficiency class 0 performance. The provisioning interface can specify up to 4 different thresholds. If the system has 5 or more class 1 cores, the 4th value is used for all remaining cores of the same class.

## Aliases and setting visibility

- **Windows Provisioning:** `HeteroIncreaseThreshold`
- **PowerCfg:** `HETEROINCREASETHRESHOLD`
- **Hidden setting:** Yes

## Values

`HeteroIncreaseThreshold` is a four-byte unsigned integer where each byte represents a threshold in percentage.

The lowest byte is the first threshold. For example, to set four thresholds—A, B, C, and D—the value of the parameter will be A + B\*256 + C\*65536 + D\*16777216.

Minimum value	0 + 0256 + 065536 + 016777216
Maximum value	100 + 100256 + 10065536 + 10016777216

## Applies to

WINDOWS EDITION	X86-BASED DEVICES	X64-BASED DEVICES	ARM-BASED DEVICES
Windows 10 for desktop editions (Home, Pro, Enterprise, and Education)	x86	amd64	N/A
Windows 10 Mobile	N/A	N/A	Supported

# HeteroDecreaseThreshold

10/2/2018 • 2 minutes to read • [Edit Online](#)

`HeteroDecreaseThreshold` specifies a threshold to cross below, which is required to park the Nth efficiency class 1 core. There is a separate value for each core index. The threshold is relative to efficiency class 0 performance. The provisioning interface can specify up to 4 different thresholds. If the system has 5 or more class 1 cores, the 4th value is used for all remaining cores of the same class.

## Aliases and setting visibility

- **Windows Provisioning:** `HeteroDecreaseThreshold`
- **PowerCfg:** `HETERODECREASETHRESHOLD`
- **Hidden setting:** Yes

## Values

`HeteroDecreaseThreshold` is a four-byte unsigned integer where each byte represents a threshold in percentage. The lowest byte is the first threshold. For example, to set four thresholds—A, B, C, and D—the value of the parameter will be  $A + B*256 + C*65536 + D*16777216$ .

Minimum value	$0 + 0256 + 065536 + 016777216$
Maximum value	$100 + 100256 + 10065536 + 10016777216$

## Applies to

WINDOWS EDITION	X86-BASED DEVICES	X64-BASED DEVICES	ARM-BASED DEVICES
Windows 10 for desktop editions (Home, Pro, Enterprise, and Education)	x86	amd64	N/A
Windows 10 Mobile	N/A	N/A	Supported

# HeteroIncreaseTime

10/2/2018 • 2 minutes to read • [Edit Online](#)

`HeteroIncreaseTime` specifies the minimum amount of time that must elapse before additional efficiency class 1 logical processors can be transitioned from the parked state to the unparked state. The time is specified in processor performance time check intervals.

## Aliases and setting visibility

- **Windows Provisioning:** `HeteroIncreaseTime`
- **PowerCfg:** `HETEROINCREASETIME`
- **Hidden setting:** Yes

## Values

The value denotes time check intervals.

Minimum value	0
Maximum value	100

## Applies to

WINDOWS EDITION	X86-BASED DEVICES	X64-BASED DEVICES	ARM-BASED DEVICES
Windows 10 for desktop editions (Home, Pro, Enterprise, and Education)	x86	amd64	N/A
Windows 10 Mobile	N/A	N/A	Supported

# HeteroDecreaseTime

10/2/2018 • 2 minutes to read • [Edit Online](#)

`HeteroDecreaseTime` specifies the minimum amount of time that must elapse before additional efficiency class 1 logical processors can be transitioned from the unparked state to the parked state. The time is specified in performance time check intervals.

## Aliases and setting visibility

- **Windows Provisioning:** `HeteroDecreaseTime`
- **PowerCfg:** `HETERODECREASETIME`
- **Hidden setting:** Yes

## Values

The value denotes time check intervals.

Minimum value	0
Maximum value	100

## Applies to

WINDOWS EDITION	X86-BASED DEVICES	X64-BASED DEVICES	ARM-BASED DEVICES
Windows 10 for desktop editions (Home, Pro, Enterprise, and Education)	x86	amd64	N/A
Windows 10 Mobile	N/A	N/A	Supported

# HeteroClass1InitialPerf

10/2/2018 • 2 minutes to read • [Edit Online](#)

`HeteroClass1InitialPerf` specifies the initial performance percentage of the efficiency class 1 core when this core is unparked.

## Aliases and setting visibility

- **Windows Provisioning:** `HeteroClass1InitialPerf`
- **PowerCfg:** `HETEROCLASS1INITIALPERF`
- **Hidden setting:** Yes

## Values

The value denotes percentage (%).

Minimum value	0
Maximum value	100

## Applies to

WINDOWS EDITION	X86-BASED DEVICES	X64-BASED DEVICES	ARM-BASED DEVICES
Windows 10 for desktop editions (Home, Pro, Enterprise, and Education)	x86	amd64	N/A
Windows 10 Mobile	N/A	N/A	Supported

# HeteroClass0FloorPerf

10/2/2018 • 2 minutes to read • [Edit Online](#)

`HeteroClass0FloorPerf` specifies the performance level floor, in percentage, to use for efficiency class 0 processors if there is at least one unparked efficiency class 1 processor.

## Aliases and setting visibility

- **Windows Provisioning:** `HeteroClass0FloorPerf`
- **PowerCfg:** `HETEROCLASS0FLOORPERF`
- **Hidden setting:** Yes

## Values

The value denotes percentage (%).

Minimum value	0
Maximum value	100

## Applies to

WINDOWS EDITION	X86-BASED DEVICES	X64-BASED DEVICES	ARM-BASED DEVICES
Windows 10 for desktop editions (Home, Pro, Enterprise, and Education)	x86	amd64	N/A
Windows 10 Mobile	N/A	N/A	Supported

# Battery settings

10/2/2018 • 2 minutes to read • [Edit Online](#)

Settings in this subgroup control the customization of battery actions and thresholds.

## Subgroup, GUID, aliases, and setting visibility

- **Subgroup:** Battery settings
- **GUID:** e73a048d-bf27-4f12-9731-8b2076e8891f
- **Windows provisioning path:** Common\Power\Policy\Settings\Battery
- **PowerCfg alias:** SUB\_BATTERY
- **Hidden setting:** Yes

## In this section

TOPIC	DESCRIPTION
Critical battery action	Specifies the action to take when the critical batter level is reached.
Critical battery threshold	Specifies a percentage of capacity when the critical battery action is taken.
Low battery action	Specifies the action to take when the low batter level is reached.
Low battery threshold	Specifies a percentage of capacity when the low battery action is taken and the <a href="#">low battery warning</a> , if enabled, appears.
Low battery warning	Specifies whether the OS displays a UI warning at the batter meter when the battery capacity crosses the low battery threshold.
Reserve battery level	Specifies a percentage of capacity when the reserve battery warning is shown to the user.

# Critical battery action

10/2/2018 • 2 minutes to read • [Edit Online](#)

Specifies the action to take when the critical batter level is reached.

## Aliases and setting visibility

- **Windows Provisioning:** `CriticalAction`
- **PowerCfg:** `BATACTIONCRIT`
- **GUID:** `637ea02f-bbcb-4015-8e2c-a1c7b9c0b546`
- **Hidden setting:** Yes

## Values

INDEX	NAME	DESCRIPTION
0	Do Nothing	No action is taken when the critical battery level is reached.
1	Sleep	The system enters sleep when the critical battery level is reached.
2	Hibernate	The system enters hibernate when the critical battery level is reached.
3	Shut Down	The system shuts down when the critical battery level is reached.

## Applies to

Available in Windows Vista and later versions of Windows.

# Critical battery threshold

10/2/2018 • 2 minutes to read • [Edit Online](#)

Specifies a percentage of capacity when the critical battery action is taken.

## Aliases and setting visibility

- **Windows Provisioning:** `CriticalBatteryLevel`
- **PowerCfg:** `BATLEVELCRIT`
- **GUID:** 9a66d8d7-4ff7-4ef9-b5a2-5a326ca2a469
- **Hidden setting:** Yes

## Values

The value denotes the percentage (%).

Minimum value	0
Maximum value	100

## Applies to

Available in Windows Vista and later versions of Windows.

# Low battery action

10/2/2018 • 2 minutes to read • [Edit Online](#)

Specifies the action to take when the low batter level is reached.

## Aliases and setting visibility

- **Windows Provisioning:** `LowAction`
- **PowerCfg:** `BATACTIONLOW`
- **GUID:** d8742dcb-3e6a-4b3c-b3fe-374623cdcf06
- **Hidden setting:** Yes

## Values

INDEX	NAME	DESCRIPTION
0	Do Nothing	No action is taken when the low battery level is reached.
1	Sleep	The system enters sleep when the low battery level is reached.
2	Hibernate	The system enters hibernate when the low battery level is reached.
3	Shut Down	The system shuts down when the low battery level is reached.

## Applies to

Available in Windows Vista and later versions of Windows.

# Low battery threshold

10/2/2018 • 2 minutes to read • [Edit Online](#)

Specifies a percentage of capacity when the low battery action is taken and the [low battery warning](#), if enabled, appears.

## Aliases and setting visibility

- **Windows Provisioning:** `LowBatteryLevel`
- **PowerCfg:** `BATLEVELLOW`
- **GUID:** 8183ba9a-e910-48da-8769-14ae6dc1170a
- **Hidden setting:** Yes

## Values

The value denotes the percentage (%).

Minimum value	0
Maximum value	100

## Applies to

Available in Windows Vista and later versions of Windows.

# Low battery warning

10/2/2018 • 2 minutes to read • [Edit Online](#)

Specifies whether the OS displays a UI warning at the batter meter when the battery capacity crosses the low battery threshold.

## Aliases and setting visibility

- **Windows Provisioning:** `LowBatteryWarning`
- **PowerCfg:** `BATFLAGSLOW`
- **GUID:** bcded951-187b-4d05-bccc-f7e51960c258
- **Hidden setting:** Yes

## Values

INDEX	NAME	DESCRIPTION
0	Disabled	The OS does not display a UI warning when the battery capacity crosses the low battery threshold.
1	Enabled	The OS displays a UI warning when the battery capacity crosses the low battery threshold.

## Applies to

Available in Windows Vista and later versions of Windows.

# Reserve battery level

10/2/2018 • 2 minutes to read • [Edit Online](#)

Specifies a percentage of capacity when the reserve battery warning is shown to the user.

## Aliases and setting visibility

- **Windows Provisioning:** ReserveBatteryLevel
- **PowerCfg:** BATLEVELRESERVE
- **GUID:** f3c5027d-cd16-4930-aa6b-90db844a8f00
- **Hidden setting:** Yes

## Values

The value denotes the percentage (%).

Minimum value	0
Maximum value	100

## Applies to

Available in Windows 7 and later versions of Windows.

# Power button and lid settings

10/2/2018 • 2 minutes to read • [Edit Online](#)

Settings in this subgroup control the customization of system button actions.

## Subgroup, GUID, aliases, and setting visibility

- **Subgroup:** Power button and lid settings
- **GUID:** 4f971e89-eebd-4455-a8de-9e59040e7347
- **Windows provisioning path:** `Common\Power\Policy\Settings\Button`
- **PowerCfg alias:** `SUB_BUTTONS`
- **Hidden setting:** Yes

## In this section

TOPIC	DESCRIPTION
<a href="#">Lid open wake action</a>	Specifies the action to take when the system lid is opened.
<a href="#">Lid switch close action</a>	Specifies the action to take when the system lid is closed.
<a href="#">Power button action</a>	Specifies the action to take when the system power button is pressed.
<a href="#">Power button forced shutdown</a>	Specifies the type of system shutdown that occurs when the system power button is pressed if the <a href="#">power button action</a> is set to Shut Down.
<a href="#">Sleep button action</a>	Specifies the action to take when the sleep power button is pressed.

# Lid open wake action

10/2/2018 • 2 minutes to read • [Edit Online](#)

Specifies the action to take when the system lid is opened.

## Aliases and setting visibility

- **Windows Provisioning:** `LidOpenWake`
- **PowerCfg:** `LIDOPENWAKE`
- **GUID:** 99ff10e7-23b1-4c07-a9d1-5c3206d741b4
- **Hidden setting:** Yes
- **Current AC power setting index:** 0x00000001
- **Current DC power setting index:** 0x00000001

## Values

INDEX	NAME	DESCRIPTION
0	Do Nothing	No action is taken when the system lid is opened.
1	Turn on the display	The OS turns on the display when the system lid is opened.

## Applies to

Available in Windows 10, version 1607 and later versions of Windows.

## Related topics

[LidNotificationsAreReliable](#)

# Lid switch close action

10/2/2018 • 2 minutes to read • [Edit Online](#)

Specifies the action to take when the system lid is closed.

## Aliases and setting visibility

- **Windows Provisioning:** `LidAction`
- **PowerCfg:** `LIDACTION`
- **GUID:** `5ca83367-6e45-459f-a27b-476b1d01c936`
- **Hidden setting:** Yes

## Values

INDEX	NAME	DESCRIPTION
0	Do Nothing	No action is taken when the system lid is closed.
1	Sleep	The system enters sleep when the system lid is closed.
2	Hibernate	The system enters hibernate when the system lid is closed.
3	Shut Down	The system shuts down when the system lid is closed.

## Applies to

Available in Windows Vista and later versions of Windows.

# Power button action

10/2/2018 • 2 minutes to read • [Edit Online](#)

Specifies the action to take when the system power button is pressed.

## Aliases and setting visibility

- **Windows Provisioning:** `PowerButtonAction`
- **PowerCfg:** `PBUTTONACTION`
- **GUID:** 7648efa3-dd9c-4e3e-b566-50f929386280
- **Hidden setting:** Yes

## Values

INDEX	NAME	DESCRIPTION
0	Do Nothing	No action is taken when the power button is pressed.
1	Sleep	The system enters sleep when the power button is pressed.
2	Hibernate	The system enters hibernate when the power button is pressed.
3	Shut Down	The system shuts down when the power button is pressed.

## Applies to

Available in Windows Vista and later versions of Windows.

# Power button forced shutdown

10/2/2018 • 2 minutes to read • [Edit Online](#)

Specifies the type of system shutdown that occurs when the system power button is pressed if the [power button action](#) is set to Shut Down.

**Warning** If you enable this setting and a user presses the power button to shut down the system, any open documents might not be saved and data loss could occur.

## Aliases and setting visibility

- **Windows Provisioning:** `ForcedShutdown`
- **PowerCfg:** `SHUTDOWN`
- **GUID:** 833a6b62-dfa4-46d1-82f8-e09e34d029d6
- **Hidden setting:** Yes

## Values

INDEX	NAME	DESCRIPTION
0	Off	A normal system shutdown will occur.
1	On	A forced system shutdown will occur.

## Applies to

Available in Windows 7 and later versions of Windows.

# Sleep button action

10/2/2018 • 2 minutes to read • [Edit Online](#)

Specifies the action to take when the sleep power button is pressed.

## Aliases and setting visibility

- **Windows Provisioning:** `SleepButtonAction`
- **PowerCfg:** `SleepButtonAction`
- **GUID:** 96996bc0-ad50-47ec-923b-6f41874dd9eb
- **Hidden setting:** Yes

## Values

INDEX	NAME	DESCRIPTION
0	Do Nothing	No action is taken when the sleep button is pressed.
1	Sleep	The system enters sleep when the sleep button is pressed.
2	Hibernate	The system enters hibernate when the sleep button is pressed.
3	Shut Down	The system shuts down when the sleep button is pressed.

## Applies to

Available in Windows Vista and later versions of Windows.

# Display settings

10/2/2018 • 2 minutes to read • [Edit Online](#)

Settings in this subgroup control the power management of the display.

## Subgroup, GUID, aliases, and setting visibility

- **Subgroup:** Display settings
- **GUID:** 7516b95f-f776-4464-8c53-06167f40cc99
- **Windows provisioning path:** Common\Power\Policy\Settings\Display
- **PowerCfg alias:** SUB\_VIDEO
- **Hidden setting:** Yes

## In this section

TOPIC	DESCRIPTION
<a href="#">Adaptive display idle timeout</a>	Specifies whether the OS automatically scales the display idle time-out based on user activity.  If the user provides input to the system shortly after the display idle timeout is reached, Windows automatically extends the display idle time-out to deliver a better user experience.
<a href="#">Allow display required policy</a>	Specifies whether Windows allows applications to temporarily prevent the display from automatically reducing brightness or turning off to save power.
<a href="#">Dim annoyance timeout</a>	This setting denotes the user annoyance detection threshold. It specifies the duration between automatic display brightness level reduction and user input to consider the automatic display brightness level reduction as an annoyance to the user.
<a href="#">Dim display brightness</a>	Denotes the reduced display brightness level after the dim idle timeout has been reached.
<a href="#">Display brightness level</a>	Specifies the default display brightness level.
<a href="#">Display idle timeout</a>	Specifies the period of inactivity before the display is automatically turned off.

# Adaptive display idle timeout

10/2/2018 • 2 minutes to read • [Edit Online](#)

Specifies whether the OS automatically scales the display idle time-out based on user activity.

If the user provides input to the system shortly after the display idle timeout is reached, Windows automatically extends the display idle time-out to deliver a better user experience.

## Aliases and setting visibility

- **Windows Provisioning:** `AdaptiveTimeout`
- **PowerCfg:** `VIDEOADAPT`
- **GUID:** 90959d22-d6a1-49b9-af93-bce885ad335b
- **Hidden setting:** Yes

## Values

INDEX	NAME	DESCRIPTION
0	Disabled	Windows does not adaptively extend the display idle timeout.
1	Enabled	Windows adaptively extends the display idle timeout.

## Applies to

Available in Windows Vista and later versions of Windows.

# Allow display required policy

10/2/2018 • 2 minutes to read • [Edit Online](#)

Specifies whether Windows allows applications to temporarily prevent the display from automatically reducing brightness or turning off to save power.

## Aliases and setting visibility

- **Windows Provisioning:** AllowDisplayRequired
- **PowerCfg:** ALLOWDISPLAY
- **GUID:** a9ceb8da-cd46-44fb-a98b-02af69de4623
- **Hidden setting:** Yes

## Values

INDEX	NAME	DESCRIPTION
0	No	Applications are not allowed to temporarily prevent display power management.
1	Yes	Applications are allowed to temporarily prevent display power management.

## Applies to

Available in Windows 7 and later versions of Windows.

# Dim annoyance timeout

10/2/2018 • 2 minutes to read • [Edit Online](#)

This setting denotes the user annoyance detection threshold. It specifies the duration between automatic display brightness level reduction and user input to consider the automatic display brightness level reduction as an annoyance to the user.

This setting applies only to portable computers that support Windows control of the brightness level of an integrated display device. In most situations, you should not change the default value of this setting.

## Aliases and setting visibility

- **Windows Provisioning:** `AdapativeIncrease`
- **PowerCfg:** `VIDEOADAPTINC`
- **GUID:** `82dbcf2d-cd67-40c5-bfdc-9f1a5ccd4663`
- **Hidden setting:** Yes

## Values

The value denotes the number of seconds.

Minimum value	0 (Do not detect user annoyance.)
Maximum value	Maximum integer

## Applies to

Available in Windows 7 and later versions of Windows.

# Dim display brightness

10/2/2018 • 2 minutes to read • [Edit Online](#)

Denotes the reduced display brightness level after the dim idle timeout has been reached.

This setting applies only to portable computers that support Windows control of the brightness level of an integrated display device.

## Aliases and setting visibility

- **Windows Provisioning:** `DimLevel`
- **PowerCfg:** `VIDEODIMLEVEL`
- **GUID:** `f1fbfdde2-a960-4165-9f88-50667911ce96`
- **Hidden setting:** Yes

## Values

The value denotes the percentage (%).

Minimum value	0
Maximum value	100

## Applies to

Available in Windows 7 and later versions of Windows.

# Display brightness level

10/2/2018 • 2 minutes to read • [Edit Online](#)

Specifies the default display brightness level.

This setting applies only to portable computers that support Windows control of the brightness level of an integrated display device.

## Aliases and setting visibility

- **Windows Provisioning:** `NormalLevel`
- **PowerCfg:** `VIDEONORMALLEVEL`
- **GUID:** `aded5e82-b909-4619-9949-f5d71dac0bcb`
- **Hidden setting:** Yes

## Values

The value denotes the percentage (%).

Minimum value	0
Maximum value	100

## Applies to

Available in Windows Vista and later versions of Windows.

# Display idle timeout

10/2/2018 • 2 minutes to read • [Edit Online](#)

Specifies the period of inactivity before the display is automatically turned off.

## Aliases and setting visibility

- **Windows Provisioning:** `IdleTimeout`
- **PowerCfg:** `VIDEOIDLE`
- **GUID:** 3c0bc021-c8a8-4e07-a973-6b14cbc2b7e
- **Hidden setting:** Yes

## Values

The value denotes the number of seconds.

Minimum value	0 (Never power off the display.)
Maximum value	Maximum integer

## Applies to

Available in Windows Vista and later versions of Windows.

# Disk settings

10/2/2018 • 2 minutes to read • [Edit Online](#)

Settings in this subgroup control the power management of disk devices.

## Subgroup, GUID, aliases, and setting visibility

- **Subgroup:** Disk settings
- **GUID:** 0012ee47-9041-4b5d-9b77-535fba8b1442
- **Windows provisioning path:** Common\Power\Policy\Settings\Disk
- **PowerCfg alias:** SUB\_DISK
- **Hidden setting:** Yes

## In this section

TOPIC	DESCRIPTION
<a href="#">Disk burst ignore time</a>	Specifies the period of inactivity to ignore when attempting to aggressively power down the disk.
<a href="#">Disk idle timeout</a>	Specifies the period of inactivity before the disk is automatically powered down.
<a href="#">Link power management mode - adaptive</a>	Specifies the period of AHCI link idle time before the link is put into a slumber state when Host-Initiated Power Management (HIPM) or Device-Initiated Power Management (DIPM) is enabled.
<a href="#">Link power management mode - HIPM/DIPM</a>	Configures the link power management mode for disk and storage devices that are attached to the system through an AHCI interface.

# Disk burst ignore time

10/2/2018 • 2 minutes to read • [Edit Online](#)

Specifies the period of inactivity to ignore when attempting to aggressively power down the disk.

## Aliases and setting visibility

- **Windows Provisioning:** N/A
- **PowerCfg:** N/A
- **GUID:** 80e3c60e-bb94-4ad8-bbe0-0d3195efc663
- **Hidden setting:** Yes

## Values

The value denotes the number of seconds.

Minimum value	0 (Do not ignore disk activity)
Maximum value	Maximum integer

## Applies to

Available in Windows Vista with Service Pack 1 (SP1), Windows Server 2008 R2, and later versions of Windows.

# Disk idle timeout

10/2/2018 • 2 minutes to read • [Edit Online](#)

Specifies the period of inactivity before the disk is automatically powered down.

## Aliases and setting visibility

- **Windows Provisioning:** `IdleTimeout`
- **PowerCfg:** `DISKIDLE`
- **GUID:** `6738e2c4-e8a5-4a42-b16a-e040e769756e`
- **Hidden setting:** Yes

## Values

The value denotes the number of seconds.

Minimum value	0 (Never idle off the disk)
Maximum value	Maximum integer

## Applies to

Available in Windows Vista and later versions of Windows.

# Link power management mode - adaptive

10/2/2018 • 2 minutes to read • [Edit Online](#)

Specifies the period of AHCI link idle time before the link is put into a slumber state when Host-Initiated Power Management (HIPM) or Device-Initiated Power Management (DIPM) is enabled.

## Aliases and setting visibility

- **Windows Provisioning:** N/A
- **PowerCfg:** N/A
- **GUID:** dab60367-53fe-4fbc-825e-521d069d2456
- **Hidden setting:** Yes

## Values

The value denotes the number of milliseconds.

Minimum value	0 (Only use partial state)
Maximum value	300,000 (5 minutes)

## Applies to

Available in Windows 7 and later versions of Windows.

# Link power management mode - HIPM/DIPM

10/2/2018 • 2 minutes to read • [Edit Online](#)

Configures the link power management mode for disk and storage devices that are attached to the system through an AHCI interface.

## Aliases and setting visibility

- **Windows Provisioning:** N/A
- **PowerCfg:** N/A
- **GUID:** 0b2d69d7-a2a1-449c-9680-f91c70521c60
- **Hidden setting:** Yes

## Values

INDEX	NAME	DESCRIPTION
0	Active	Link power management is not used.
1	HIPM	Host-Initiated Power Management (HIPM) is used.
2	HIPM and DIPM	HIPM and Device-Initiated Power Management (DIPM) are used.

## Applies to

Available in Windows 7 and later versions of Windows.

# Energy Saver settings

10/2/2018 • 2 minutes to read • [Edit Online](#)

Settings in this subgroup control the battery threshold and brightness when Energy Saver is turned on.

## Subgroup, GUID, aliases, and setting visibility

- **Subgroup:** Energy Saver settings
- **GUID:** de830923-a562-41af-a086-e3a2c6bad2da
- **Windows provisioning path:** Common\Power\Policy\Settings\EnergySaver
- **PowerCfg alias:** SUB\_ENERGYSAYER
- **Hidden setting:** Yes

## In this section

TOPIC	DESCRIPTION
<a href="#">Battery threshold</a>	Specifies the battery charge level, as a percentage, at which Energy Saver is turned on.
<a href="#">Brightness</a>	Specifies the percentage value to scale brightness to when Energy Saver is turned on.

# Battery threshold

10/2/2018 • 2 minutes to read • [Edit Online](#)

Specifies the battery charge level, as a percentage, at which Energy Saver is turned on.

## Aliases and setting visibility

- **Windows Provisioning:** `BatteryThreshold`
- **PowerCfg:** `ESBATTTHRESHOLD`
- **GUID:** `e69653ca-cf7f-4f05-aa73-cb833fa90ad4`
- **Hidden setting:** Yes

## Values

The value denotes the percentage (%).

Minimum value	0
Maximum value	100

## Applies to

Available in Windows 10 and later versions of Windows.

# Brightness

10/2/2018 • 2 minutes to read • [Edit Online](#)

Specifies the percentage value to scale brightness to when Energy Saver is turned on.

## Aliases and setting visibility

- **Windows Provisioning:** Brightness
- **PowerCfg:** ESBRIGHTNESS
- **GUID:** 13d09884-f74e-474a-a852-b6bde8ad03a8
- **Hidden setting:** Yes

## Values

The value denotes the percentage (%).

Minimum value	0
Maximum value	100

## Applies to

Available in Windows 10 and later versions of Windows.

# PCI Express settings

10/2/2018 • 2 minutes to read • [Edit Online](#)

Settings in this subgroup control the power management of PCI Express links.

## Subgroup, GUID, aliases, and setting visibility

- **Subgroup:** PCI Express settings
- **GUID:** 501a4d13-42af-4429-9fd1-a8218c268e20
- **Windows provisioning path:** Common\Power\Policy\Settings\PCIEexpress
- **PowerCfg alias:** SUB\_PCIEPRESS
- **Hidden setting:** Yes

## In this section

TOPIC	DESCRIPTION
<a href="#">Link state power management</a>	Specifies the personality of the power plan.

# Link state power management

10/2/2018 • 2 minutes to read • [Edit Online](#)

Specifies the personality of the power plan.

**Warning** System administrators should not change the power plan personality settings.

## Aliases and setting visibility

- **Windows Provisioning:** `ASPM`
- **PowerCfg:** `ASPM`
- **GUID:** ee12f906-d277-404b-b6da-e5fa1a576df5
- **Hidden setting:** Yes

## Values

INDEX	NAME	DESCRIPTION
0	None	The power plan is a Power Saver plan.
1	Moderate Power Savings	The system attempts to use the L0 state when the link is idle.
2	Maximum Power Savings	The system attempts to use the L1 state when the link is idle.

## Applies to

Available in Windows Vista and later versions of Windows.

# Sleep settings

10/2/2018 • 2 minutes to read • [Edit Online](#)

Settings in this subgroup control sleep, resume, and other related functionality.

## Subgroup, GUID, aliases, and setting visibility

- **Subgroup:** Sleep settings
- **GUID:** 238c9fa8-0aad-41ed-83f4-97be242c8f20
- **Windows provisioning path:** Common\Power\Policy\Settings\Sleep
- **PowerCfg alias:** SUB\_SLEEP
- **Hidden setting:** Yes

## In this section

TOPIC	DESCRIPTION
Allow away mode	Specifies whether the system uses away mode. If this setting is disabled, away mode is not used even if programs request it.
Allow sleep with open remote files	Configures the network file system to prevent the computer from automatically entering sleep when remote network files are open.
Allow sleep states	Specifies whether the system uses low power sleep states.
Allow system required requests	Configures the power manager to accept or ignore application system required requests. These requests prevent the system from automatically entering sleep after a period of user inactivity.
Automatically wake for tasks	Specifies whether the system uses the system-wide wake-on-timer capability.  The system can automatically use wake-on-timer on capable hardware to perform scheduled tasks. For example, the system might wake automatically to install updates.
Hibernate idle timeout	Specifies the duration of time after sleep that the system automatically wakes and enters hibernation.
Hybrid sleep	Specifies whether the system can enter hybrid sleep.

TOPIC	DESCRIPTION
<a href="#">Sleep idle timeout</a>	Specifies the duration of inactivity before the system automatically enters sleep.
<a href="#">Sleep unattended idle timeout</a>	Specifies the duration of inactivity before the system automatically enters sleep after waking from sleep in an unattended state.

# Allow away mode

10/2/2018 • 2 minutes to read • [Edit Online](#)

Specifies whether the system uses away mode. If this setting is disabled, away mode is not used even if programs request it.

## Aliases and setting visibility

- **Windows Provisioning:** `AwayMode`
- **PowerCfg:** `AWAYMODE`
- **GUID:** 25dfa149-5dd1-4736-b5ab-e8a37b5b8187
- **Hidden setting:** Yes

## Values

INDEX	NAME	DESCRIPTION
0	Disabled	Away mode is not available.
1	Enabled	Away mode is available.

## Applies to

Available in Windows Vista and later versions of Windows.

# Allow sleep with open remote files

10/2/2018 • 2 minutes to read • [Edit Online](#)

Configures the network file system to prevent the computer from automatically entering sleep when remote network files are open.

## Aliases and setting visibility

- **Windows Provisioning:** AllowRemoteOpenSleep
- **PowerCfg:** ALLOWREMOTEOPENSLEEP
- **GUID:** d4c1d4c8-d5cc-43d3-b83e-fc51215cb04d
- **Hidden setting:** Yes

## Values

INDEX	NAME	DESCRIPTION
0	Off	Prevents automatic sleep when remote network files are open. However, if the open files are stored in Offline Files and are backed by the Offline File cache, automatic sleep is allowed.
1	On	Prevents automatic sleep when remote network files are open. However, if the open files are stored in Offline Files or the open files have not been updated since they were originally opened, automatic sleep is allowed.

## Applies to

Available in Windows Vista and later versions of Windows.

# Allow sleep states

10/2/2018 • 2 minutes to read • [Edit Online](#)

Specifies whether the system uses low power sleep states.

## Aliases and setting visibility

- **Windows Provisioning:** AllowStandby
- **PowerCfg:** ALLOWSTANDBY
- **GUID:** abfc2519-3608-4c2a-94ea-171b0ed546ab
- **Hidden setting:** Yes

## Values

INDEX	NAME	DESCRIPTION
0	Disabled	Sleep states (ACPI S1, S2, and S3) are not available.
1	Enabled	Sleep states (ACPI S1, S2, and S3) are available.

## Applies to

Available in Windows Vista and later versions of Windows.

# Allow system required requests

10/2/2018 • 2 minutes to read • [Edit Online](#)

Configures the power manager to accept or ignore application system required requests. These requests prevent the system from automatically entering sleep after a period of user inactivity.

## Aliases and setting visibility

- **Windows Provisioning:** `AllowSystemRequired`
- **PowerCfg:** `SYSTEMREQUIRED`
- **GUID:** a4b195f5-8225-47d8-8012-9d41369786e2
- **Hidden setting:** Yes

## Values

INDEX	NAME	DESCRIPTION
0	No	Application system required requests will be ignored.
1	Yes	Application system required requests will be accepted.

## Applies to

Available in Windows 7 and later versions of Windows.

# Automatically wake for tasks

10/2/2018 • 2 minutes to read • [Edit Online](#)

Specifies whether the system uses the system-wide wake-on-timer capability.

The system can automatically use wake-on-timer on capable hardware to perform scheduled tasks. For example, the system might wake automatically to install updates.

## Aliases and setting visibility

- **Windows Provisioning:** `AllowRtcWake`
- **PowerCfg:** `RTCWAKE`
- **GUID:** bd3b718a-0680-4d9d-8ab2-e1d2b4ac806d
- **Hidden setting:** Yes

## Values

INDEX	NAME	DESCRIPTION
0	No	Wake on timer is disabled.
1	Yes	Wake on timer is enabled.
2	Important	Wake on internal system timers only.

## Applies to

Available in Windows Vista and later versions of Windows.

# Hibernate idle timeout

10/2/2018 • 2 minutes to read • [Edit Online](#)

Specifies the duration of time after sleep that the system automatically wakes and enters hibernation.

This setting enables hibernate option on Modern Standby systems. Set the value to 0 to disable the feature.

## Aliases and setting visibility

- **Windows Provisioning:** `HibernateTimeout`
- **PowerCfg:** `HIBERNATEIDLE`
- **GUID:** `9d7815a6-7ee4-497e-8888-515a05f02364`
- **Hidden setting:** Yes

## Values

The value denotes the number of seconds.

Minimum value	0 (Never idle to sleep)
Maximum value	Maximum integer

## Applies to

Available in Windows Vista and later versions of Windows.

# Hybrid sleep

10/2/2018 • 2 minutes to read • [Edit Online](#)

Specifies whether the system can enter hybrid sleep.

## Aliases and setting visibility

- **Windows Provisioning:** `HybridSleep`
- **PowerCfg:** `HYBRIDSLEEP`
- **GUID:** `94ac6d29-73ce-41a6-809f-6363ba21b47e`
- **Hidden setting:** Yes

## Values

INDEX	NAME	DESCRIPTION
0	Disabled	Hybrid sleep is disabled.
1	Enabled	Hybrid sleep is enabled.

## Applies to

Available in Windows Vista and later versions of Windows.

# Sleep idle timeout

10/2/2018 • 2 minutes to read • [Edit Online](#)

Specifies the duration of inactivity before the system automatically enters sleep.

## Aliases and setting visibility

- **Windows Provisioning:** StandbyTimeout
- **PowerCfg:** STANDBYIDLE
- **GUID:** 29f6c1db-86da-48c5-9fdb-f2b67b1f44da
- **Hidden setting:** Yes

## Values

The value denotes the number of seconds.

Minimum value	0 (Never idle to sleep)
Maximum value	Maximum integer

## Applies to

Available in Windows Vista and later versions of Windows.

# Sleep unattended idle timeout

10/2/2018 • 2 minutes to read • [Edit Online](#)

Specifies the duration of inactivity before the system automatically enters sleep after waking from sleep in an unattended state.

For example, if the system wakes from sleep because of a timed event or a wake on LAN (WoL) event, the sleep unattended idle timeout value will be used instead of the [sleep idle timeout](#) value.

## Aliases and setting visibility

- **Windows Provisioning:** `UnattendTimeout`
- **PowerCfg:** `UnattendTimeout`
- **GUID:** 7bc4a2f9-d8fc-4469-b07b-33eb785aaca0
- **Hidden setting:** Yes

## Values

The value denotes the number of seconds.

Minimum value	0 (Never idle to sleep)
Maximum value	Maximum integer

## Applies to

Available in Windows Vista with Service Pack 1 (SP1), Windows Server 2008 R2, and later versions of Windows.

# Other power settings

10/2/2018 • 2 minutes to read • [Edit Online](#)

Settings in this subgroup do not belong to any other subgroup.

## Subgroup, GUID, aliases, and setting visibility

- **Subgroup:** No subgroup settings
- **GUID:** fea3413e-7e05-4911-9a71-700331f1c294
- **Windows provisioning path:** Common\Power\Policy\Settings\Misc
- **PowerCfg alias:** SUB\_NONE
- **Hidden setting:** Yes

## In this section

TOPIC	DESCRIPTION
<a href="#">Device idle policy</a>	Determines whether conservation idle timeouts or performance idle timeouts are used for devices that are integrated with Windows kernel power manager device idle detection.
<a href="#">Prompt for password on resume</a>	Specifies whether the user must enter a password at the secure desktop when the system resumes from sleep.  <div style="border: 1px solid black; padding: 5px;"><b>Note</b> All Windows desktop editions have this setting enabled by default. This is a change from Windows 8.1 and earlier which had the setting disabled by default on some editions.</div>
<a href="#">Allow networking during standby</a>	Specifies whether to allow networking during standby.

# Device idle policy

10/2/2018 • 2 minutes to read • [Edit Online](#)

Determines whether conservation idle timeouts or performance idle timeouts are used for devices that are integrated with Windows kernel power manager device idle detection.

## Aliases and setting visibility

- **Windows Provisioning:** `DeviceIdlePolicy`
- **PowerCfg:** `N/A`
- **GUID:** 4faab71a-92e5-4726-b531-224559672d19
- **Hidden setting:** Yes

## Values

INDEX	NAME	DESCRIPTION
0	Performance	Power idle timeouts are used.
1	Power Savings	Conservation idle timeouts are used.

## Applies to

Available in Windows Vista with Service Pack 1 (SP1), Windows Server 2008 R2, and later versions of Windows.

# Prompt for password on resume

10/2/2018 • 2 minutes to read • [Edit Online](#)

Specifies whether the user must enter a password at the secure desktop when the system resumes from sleep.

**Note** All Windows desktop editions have this setting enabled by default. This is a change from Windows 8.1 and earlier which had the setting disabled by default on some editions.

## Aliases and setting visibility

- **Windows Provisioning:** `LockConsoleOnWake`
- **PowerCfg:** `CONSOLELOCK`
- **GUID:** `0e796bdb-100d-47d6-a2d5-f7d2daa51f51`
- **Hidden setting:** Yes

## Values

INDEX	NAME	DESCRIPTION
0	Disabled	The system returns to the desktop when resuming from sleep.
1	Enabled	The system returns to the secure desktop, and the user must enter a password when the system resumes from sleep.

## Applies to

Available in Windows Vista and later versions of Windows.

# Allow networking during standby

10/2/2018 • 2 minutes to read • [Edit Online](#)

Specifies whether to allow networking during standby.

## Aliases and setting visibility

- **Windows Provisioning:** `ConnectivityInStandby`
- **GUID:** f15576e8-98b7-4186-b944-eafa664402d9
- **Hidden setting:** Yes

## Values

INDEX	NAME	DESCRIPTION
0	Disabled	The system will disconnect from the network during standby.
1	Enabled	The system will stay connected to the network during standby.
2	Managed by Windows	Windows will manage network connectivity during standby.

## Applies to

Available in Windows Vista and later versions of Windows.

# Legacy configuration options

10/2/2018 • 2 minutes to read • [Edit Online](#)

The processor power settings documented in this section are no longer supported for platform configuration. However, system administrators and power users may use them.

## Options for performance state engine

You can use the following options to configure the performance state engine:

- [PERFBOOSTMODE](#)
- [PERFBOOSTPOL](#)

# PERFBOOSTMODE

10/2/2018 • 2 minutes to read • [Edit Online](#)

**PERFBOOSTMODE** determines how processors select a performance level when current operating conditions allow for boosting performance above the nominal level.

## GUID, alias, and setting visibility

- **GUID:** be337238-0d82-4146-a960-4f3749d470c7
- **PowerCfg alias:** `PERFBOOSTMODE`
- **Hidden setting:** Yes

## Values

INDEX	NAME
0	Disabled
1	Enabled
2	Aggressive
3	Efficient enabled
4	Efficient aggressive

## Applies to

WINDOWS EDITION	X86-BASED DEVICES	X64-BASED DEVICES	ARM-BASED DEVICES
Windows 10 for desktop editions (Home, Pro, Enterprise, and Education)	x86	amd64	N/A
Windows 10 Mobile	N/A	N/A	Supported

# PERFBOOSTPOL

10/2/2018 • 2 minutes to read • [Edit Online](#)

`PERFBOOSTPOL` configures the processor performance boost policy.

## GUID, alias, and setting visibility

- **GUID:** 45bcc044-d885-43e2-8605-ee0ec6e96b59
- **PowerCfg alias:** `PERFBOOSTPOL`
- **Hidden setting:** Yes

## Values

The value denotes percentage (%).

Minimum value	0
Maximum value	100

## Applies to

WINDOWS EDITION	X86-BASED DEVICES	X64-BASED DEVICES	ARM-BASED DEVICES
Windows 10 for desktop editions (Home, Pro, Enterprise, and Education)	x86	amd64	N/A
Windows 10 Mobile	N/A	N/A	Supported

# Preinstalled and exclusive apps

10/2/2018 • 2 minutes to read • [Edit Online](#)

As an OEM, you have a unique opportunity to create applications that ship with your OS image directly to customers. This means you can preinstall applications onto the image, connect them to devices, and promote them in both the Microsoft Store, and your OEM store. You can also promote exclusive apps in your OEM store.

## App design

To make a compelling app that gets your customers to pay attention, follow the design principles that guide the development of great Universal Windows Platform (UWP) experiences. The [Introduction to UWP app design](#) is a great starting place for learning about UWP. From there you should learn about the [controls and control patterns](#) to use, how to interact with [inputs and devices](#), and how to think about [usability](#). The [Get Started with Windows Apps](#) guide in the [Windows Dev Center](#) is another resource you can use to learn more.

## Preinstalled apps

The primary channel for distributing apps is the Microsoft Store. However, because Microsoft Store apps are only available on the device after a user-initiated download and some partner apps need to be available at first boot, there is an alternate option available for OEMs and mobile operators. OEMs and Mobile operators can create Partner applications that can be packaged and configured to install during the initial device setup process. While the user is going through the initial setup process, the preinstalled applications are installed in the background.

## Exclusive apps

By forming relationships with developers, you can work together to publish exclusive apps onto your devices. You have the flexibility based on the contracts you establish with developers to ensure that these apps are available exclusively on your Windows 10-based devices (as identified by OEM Store ID) and don't appear in the general catalog on any other devices.

## In this section

TOPIC	DESCRIPTION
<a href="#">Preinstallable apps for desktop devices</a>	Learn how to add an app to a Windows 10 for desktop editions (Home, Pro, Enterprise, and Education) image that will be available to customers at first boot.
<a href="#">Preinstallable apps for mobile devices</a>	Learn how to add an app to a mobile image that will be available to customers at first boot.
<a href="#">Preinstall tasks</a>	OEMs and MOs are permitted to ship preinstalled apps in the device image. Some of those preinstalled apps require tasks to run without user interaction and often before the end-user opens the app for the first time; such as a product survey app or a SMS server registration. Similarly, some apps will need servicing tasks to run without user interaction after an app has been updated. Preinstall and update tasks provide the mechanism for allowing tasks to run in the background without before the app is installed or when it is updated.

TOPIC	DESCRIPTION
<a href="#">Exclusive apps</a>	Learn how to set the OEM Store ID and SCM ID in the registry to enable exclusive apps for your devices.

## Audience

Preinstalled and exclusive app guidance is designed for use by OEM and MO developers.

# Exclusive apps

10/2/2018 • 4 minutes to read • [Edit Online](#)

By forming relationships with developers, you can work together to publish exclusive apps onto your devices. You have the flexibility based on the contracts you establish with developers to ensure that these apps are available exclusively on your Windows 10-based devices (as identified by OEM Store ID) and don't appear in the general catalog on any other devices.

Your exclusive apps must be submitted to the Microsoft Store by a developer that you are in an exclusive relationship with, as defined in the Microsoft Store OEM Program enrollment form. You may also set up your own exclusive developer account where apps published from that account are automatically designated as exclusive.

## NOTE

Exclusive app accounts are only available to OEM developer accounts unless you have been granted an exception.

## Set your OEM Store ID and SCM ID

You must insert a combination of unique identifiers into your images to facilitate exclusive app features. These features rely on the OEM Store ID (previously referred to as the OEM ID) and optionally the Store Content Modifier ID (SCM ID). The OEM Store ID and the Store Content Modifier (SCM) will be provided by the Microsoft Store Partner Operations team at your request.

### OEM Store ID

The OEM Store ID is required for you to create your OEM Store in the Microsoft Store and for app exclusivity. Here are the characteristics of an OEM Store ID:

- Assignment is a one-time operation; each OEM should have one and only one unique OEM Store ID.
- You must populate your OEM Store ID to the registry field specified in the [Registry Field Requirements](#) section below.
- OEM Store ID is set during manufacturing and cannot be changed later. For manufacturing guidance, see [OEM Deployment of Windows 10 for desktop editions](#).

### SCM ID

The Store Content Modifier ID (SCM ID) is optional. You can use it in addition to your OEM Store ID, but never instead of the OEM Store ID. Here are the characteristics of an SCM ID:

- SCM IDs are used to differentiate between different sub-brands, devices, and/or device groups belonging to a given OEM for specialized OEM Store or exclusive app experiences.
- Each OEM can have multiple SCM IDs; however, each specified segment (devices/brands/devices groups) must be identified by a single, unique SCM ID populated on all devices belonging to the given target segment.
- Each device can have at most one SCM ID.
- You populate the SCM ID to the registry field specified in the [Registry Field Requirements](#) section below.
- The SCM ID is set during manufacturing and cannot be changed later. For manufacturing guidance, see [OEM Deployment of Windows 10 for desktop editions](#).
- Depending on your business needs and objectives, you decide what level of granularity you want to use for your SCM IDs. However, consider that additional SCM IDs add more complexity and will take more work and resources to support. The table below describes some possible options.

OEM-ID	SCM ID	Description
FABRIKAM	Fabrikam_Enterprise	Device group - segmenting devices geared towards enterprise customers.
FABRIKAM	Fabrikam_Proseware	Brand group - grouping of the "Proseware" branded devices.
FABRIKAM	Fabrikam_ProsewareX2	Device group - device-specific segment, Fabrikam Proseware model X2.

#### NOTE

If you do not currently have an SCM or chose not to differentiate your OEM Store by using a SCM, contact the Microsoft Store Partner Operations Team for instructions on setting up an OEM Store using only an OEM Store ID. If you have an SCM ID that was created for your Windows 8.1 devices, you can still use that SCM ID for your Windows 10-based device. If you do so, you will still need to obtain your OEM Store ID from the Microsoft Store Partner Operations Team and insert that into your Windows 10 image.

### Registry field requirements

In order to use exclusive app features, you must set the following fields in the registries of the target devices.

Key	Location	Type	Value
OEMID	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Store	REG_SZ	The OEM Store ID provided by the Microsoft Store Partner Operations team.
StoreContentModifier	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Store	REG_SZ	The SCM ID provided by the Microsoft Store Partner Operations team.

#### NOTE

The **HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Store\Configuration** key should not be used by OEMs. It is where the Store application writes the values it discovers so that the OS components are using the same settings cross applications.

If you are using Windows Configuration Designer to create an image, you can add a Windows setting:

`WindowsStore\ContentModifier`

### Microsoft Store process

The following list shows some of the key tasks and workflows for engaging with the Store. Depending on your needs, the order and importance of each of these tasks and workflows varies:

- OEM works with Microsoft Store Partner Operations on plans for their exclusive Store in Store.
- Microsoft Store Partner Operations creates device identification marker for inclusion in the registry.
- OEM builds device ID into the registry.
- Microsoft Store Partner Operations enables OEM Store merchandising tools.
- OEM curates content and controls publishing of OEM Store content.

## Send SMBIOS information to Microsoft

Because the OEM Store ID is not saved upon migration, including upon upgrade, you must ensure that Microsoft has the correct SMBIOS details of the device to enable it for customized Store experiences. Please contact [partnerops@microsoft.com](mailto:partnerops@microsoft.com) with the **SMBIOS Manufacturer**, and **SMBIOS Product Name**, for your device.

- The SMBIOS Manufacturer value can be found under the registry key  
`HKLM\System\CurrentControlSet\Control\SystemInformation\Manufacturer`
- The SMBIOS Product Name can be found under the registry key  
`HKLM\System\CurrentControlSet\Control\SystemInformation\SystemProductName`

# Preinstallable apps for desktop devices

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs and Mobile operators can create Partner applications that can be packaged and configured to install during the initial device setup process. While the user is going through the initial setup process, the preinstalled applications are installed in the background.

The process for creating a preinstalled app is similar to that of a standard app. An unsigned app package (.appx), generated with the Windows SDK, is submitted to the Windows Dev Center for certification and signing. During the submission process, you can specify that you are submitting a preinstalled app. If the app meets certification requirements, it is processed to create a package that can be downloaded from the Dev Center. The app can then be published to the Microsoft Store as well, so that users who have uninstalled the app can re-download it and updates can later be offered to devices that have the app installed.

Some characteristics of preinstalled apps include:

1. They can be published as "hidden" so that the app is not discoverable in the Microsoft Store except through a deep link.
2. They can be updated, as live or hidden to the Microsoft Store. Users with the preinstalled application will get a notification for the update.
3. They can be deleted by the user. They can be reinstalled if published live.
4. They can become obsolete. If a user uninstalls an app that is no longer sold in the Microsoft Store, the user will not be able to reinstall that app.

## Get the Windows ADK

You will need to use the Windows Assessment and Deployment Kit (ADK) to pre-install Microsoft Store apps in your desktop images. [Download the Windows ADK](#).

## Request a preinstallation package

Once an app has been added to the Dev Center, you can request a preinstallation package for it. If you are the OEM adding this application to your OS image, you would ask the developer of the application to do this on your behalf. They would then give you the downloaded zip file. You cannot access their developer account directly.

1. From the dashboard in Dev Center, select the app that is to be preinstalled. If it is a new app, click **Create new app**.
2. Select **manage published** packages
3. Select **Request package** for OS preinstallation
4. A confirmation dialog will appear, noting that apps preinstalled on an OS prior to Windows 10 must be free. Select **Enable**.
5. Find the correct package for the targeted OS and download by selecting **Download** or **Generate package**.
6. Once ready the link will change to **Download**.
7. Zip file is ready for inclusion in OS image.

## Add the app to the OS image

Applications are considered Assets, which are configurable customizations that are not settings. You can add them using DISM, which is part of the Windows Assessment and Deployment Kit (ADK). In Windows 10, version 1803, you can use DISM to provision apps per region.

For detailed instructions, see [Preinstall apps using DISM](#).

# Preinstallable apps for mobile devices

10/2/2018 • 5 minutes to read • [Edit Online](#)

## To add a preinstalled app to a mobile image

The process for creating a preinstallable app is similar to that of a standard app. In the Windows 10 Dev Center, a developer submits an app that you want to preinstall on your Windows 10 Mobile image. Once the app is submitted, you can request a preinstallation package, download it, and add it to the image, as described in this topic.

To add a preinstallable app, you will need to perform the following actions:

- Request a preinstallation package
- Create a .provxml for the preinstallable app
- Add the app to the image with Customization answer file
- Build the image

For more information about customization answer files, see [Customization answer file](#). For more information about building with Customization answer files, see [Building a mobile image using ImgGen.cmd](#).

## Request a preinstallation package

Developers who have added an app to the Dev Center can request a preinstallation package for it. They can then give the preinstallation package directly to the OEM they are working with. If you are the OEM adding this application to your OS image, you would ask the developer of the application to download the application package and then give you the downloaded zip file. You cannot access their developer account directly. Once you have the preinstall package, you can continue with the rest of the steps. For more information on how a developer generates preinstall packages for an OEM, see [Generate preinstall packages for OEMs](#).

## Create a .provxml file for a preinstallable app

Adding a preinstalled app to an Windows 10 Mobile OS image requires a .provxml configuration file that specifies the installation parameters and the Windows 10 Store catalog identifiers. Specifically, it should specify the path to the .appx file, the path to the license file, and the Store catalog IDs. This information is used when the app connects to the Store service to check for updates. To minimize the chance of error, the developer portal provides the appropriate XML for your app. The following is an example of what the .provxml might look like.

```

<?xml version="1.0" encoding="UTF-8" ?>
<wap-provisioningdoc>
    <characteristic type="AppInstall">
        <characteristic type="AppXPackage">
            <parm name="ProductID" value="{09f2d20a-7076-4970-80ac-1bc24c171d2e}" />
            <parm name="AppXPath" value="c:\Programs\CommonFiles\Xaps\SampleApp.appx"/>
            <parm name="LicensePath" value="c:\Programs\CommonFiles\Xaps\SampleAppLicense.xml"/>
            <parm name="InstanceID" value="{03e9a435-3000-11db-89ca-0019b92FFFFF}" />
            <parm name="OfferID" value="{03e9a435-3000-11db-89ca-0019b92FFFFF}" />
            <parm name="PayloadID" value="{03e9a435-3000-11db-89ca-0019b92FFFFF}" />
            <parm name="UninstallDisabled" value="false" />
            <parm name="FullyPreInstall" value="false" />
            <parm name="ForceUpdate" value="false" />
        </characteristic>
    </characteristic>
</wap-provisioningdoc>

```

#### **NOTE**

provxml files for preinstalled apps must follow a prescribed naming convention. You must use MPAP\_name\_index.provxml, where name and index can be any strings. Typically, name is the name of the update package that contains the preinstalled app, and index is a string that differentiates provxml files that have the same name. Often, index is represented as a number, such as 01.

#### **provxml flags**

These are the flags you can use in your provxml.

FLAG	DESCRIPTION
UninstallDisabled	This flag controls whether a preinstalled app can be uninstalled by a user. When set to FALSE(default), a user is able to uninstall the preinstalled app. When set to TRUE, a user is not able to uninstall the app. This flag is only settable via provxml and cannot be overridden through a Store update. Only a device update with an updated provxml file can change this value. Ideally, to maintain the user experience, this flag should only be set to TRUE for apps that are critical to phone functionality.
ForceUpdate	This flag allows an app in an OS update image to attempt to overwrite an existing version of the app already installed on the phone prior to update to Windows 10 Mobile. The default value for this flag is FALSE. Be aware that because the app update is forced, setting this flag to TRUE might result in a downgrade in functionality if the already-installed app was developed for an earlier version of the OS. In general, this flag should only be used when the Windows 10 Mobile version of the app must be on the phone immediately after update, even if it means downgrading the version of the app already installed.

FLAG	DESCRIPTION
FullyPreinstall	<p>This flag controls whether the app is MDIL bound during first boot/update or whether it is delayed until after those operations complete. Delaying MDIL binding, which is the default behavior for apps that are not pre-pinned to Start, allows the user to get back to their phone as quickly as possible. When binding is deferred till after first boot/update completes the app icon will display greyed out with a status of "installing" and cannot be run until the deferred bind completes. The amount of time it takes to complete all deferred bindings is dependent on the number of deferred preloaded apps and the user's activity. The flag behavior is as follows:</p> <ul style="list-style-type: none"> <li>• <b>true</b>: MDIL binding occurs before first boot or update completes.</li> <li>• <b>false</b>: If the app is pre-pinned to Start, MDIL binding is performed before first boot or update completes. If the app is not pre-pinned to Start, MDIL binding is deferred until after first boot or update completes.</li> </ul> <p>Generally, this value should be left as the default (FALSE) unless the app must be available to run immediately after first boot or an OS update. Some example situations where this flag should be set to TRUE are the following:</p> <ul style="list-style-type: none"> <li>• OEM extension apps</li> <li>• Phone dialer-installed apps</li> <li>• OEM service agents</li> <li>• Critical system settings apps</li> </ul>

## Add the app to the image

Preinstalling apps are added to the OS image using a customizations.xml answer file. To create the customizations.xml answer file, first [install the Windows Configuration Designer](#), and then [create a provisioning package](#). You can then open the project folder to find the customizations.xml file.

To include preinstalled apps in your image, you must add the `<Application>` element to your customizations.xml file with the appropriate defining attributes. The following code sample illustrates how an app would be added to a customization answer file for preinstalling.

```

<Applications>
  <Application
    License="$(CAFE_OUTPUT_DIR)\content\App_MobileTV_7e7cc86e_e1c0_476a_ac88_db3c9ffffabb\MobileTV_License.xml"
    ProvXML="$(CAFE_OUTPUT_DIR)\content\App_MobileTV_7e7cc86e_e1c0_476a_ac88_db3c9ffffabb\MPAP_MobileTV_01.provxml"
    Source="$(CAFE_OUTPUT_DIR)\content\App_MobileTV_7e7cc86e_e1c0_476a_ac88_db3c9ffffabb\MobileTV.xap"/>
  <Application
    License="$(CAFE_OUTPUT_DIR)\content\App_AudioSettings_373cb76e_7f6c_45aa_8633_b00e85c73261\audio_License.xml"
    ProvXML="$(CAFE_OUTPUT_DIR)\content\App_AudioSettings_373cb76e_7f6c_45aa_8633_b00e85c73261\MPAP_audio_01.provxml"
    Source="$(CAFE_OUTPUT_DIR)\content\App_AudioSettings_373cb76e_7f6c_45aa_8633_b00e85c73261\audio.appx"/>
  <Application
    License="$(CAFE_OUTPUT_DIR)\content\App_MicrosoftHealthApp_0168b504_ca18_46b8_b60a_0f6fdc271c81\MicrosoftHealthApp_License.xml"
    ProvXML="$(CAFE_OUTPUT_DIR)\content\App_MicrosoftHealthApp_0168b504_ca18_46b8_b60a_0f6fdc271c81\MPAP_MicrosoftHealthApp_01.provxml"
    Source="$(CAFE_OUTPUT_DIR)\content\App_MicrosoftHealthApp_0168b504_ca18_46b8_b60a_0f6fdc271c81\MicrosoftHealthApp.appxbundle"/>
</Applications>

```

**NOTE**

The provxml file must be placed in the "\$(runtime.commonfiles)\Provisioning\OEM" directory. The license file and app package (.xap or .appx) must be placed in the "\$(runtime.commonfiles)\xaps" directory

After you've configured your customizations.xml answer file, build the image using the Windows Configuration Designer command-line interface. See [Windows Configuration Designer command-line interface](#) for instructions.

## Build the image

Follow the steps in the [Build a customized mobile image using imggen](#)

# Preinstall tasks

10/2/2018 • 2 minutes to read • [Edit Online](#)

OEMs and MOs are permitted to ship preinstalled apps in the device image. Some of those preinstalled apps require tasks to run without user interaction and often before the end-user opens the app for the first time; such as a product survey app or a SMS server registration. Similarly, some apps will need servicing tasks to run without user interaction after an app has been updated. Preinstall and update tasks provide the mechanism for allowing tasks to run in the background without before the app is installed or when it is updated.

There are two deployments task types available to UAPs: PreInstallConfigTask and UpdateTask. Both are IBackgroundTasks.

Here are the general rules that govern these tasks.

- Your app manifest can contain only one PreInstallConfigTask and one UpdateTask.
- Deployment tasks are applicable to any platform type.
- Deployment tasks can execute after the deployment operation has been completed and committed.
- Failed deployment tasks are not restarted.
- Failed deployment tasks do not affect the successful deployment of the app.
- Deployment tasks are not restarted after reboot.
- Deployment tasks should not depend on one another.

## Code Examples

### **UpdateTask example**

Update task is supported for any possible update path, for example:

- .xap to .xap
- .xap to .appx
- .xap to .appxbundle
- .appx to .appx
- .appx to .appxbundle
- .appxbundle to .appxbundle

Here's the example .appx manifest:

```

<Package>
  <Extensions>
    <Extension Category="windows.activatableClass.inProcessServer">
      <InProcessServer>
        <Path>App.dll</Path>
        <ActivatableClass ActivatableClassId="App.UpdateTask" ThreadingModel="MTA"/>
      </InProcessServer>
    </Extension>
  </Extensions>

  <Applications>
    <Application>
      <Extensions>
        <Extension Category="windows.updateTask" EntryPoint="App.UpdateTask">
        </Extension>
      </Extensions>
    </Application>
  </Applications>
</Package>

```

Here's the example C# code:

```

public sealed class UpdateTask : IBackgroundTask
{
    public async void Run(IBackgroundTaskInstance taskInstance)
    {
        CancellationTokenSource cts = new CancellationTokenSource();
        var deferral = taskInstance.GetDeferral();
        taskInstance.Canceled += 
            (sender, reason) =>
        {
            cts.Cancel();
        };
        try
        {
            await MigrateApp(); // Do app migration/update steps.
        }
        catch (TaskCanceledException x)
        {
            // do nothing on cancellation.
        }
        deferral.Complete();
    }
}

```

### **PreInstallConfig Task task example**

Here's the example .appx manifest:

```

<Package>
  <Extensions>
    <Extension Category="windows.activatableClass.inProcessServer">
      <InProcessServer>
        <Path>App.dll</Path>
        <ActivatableClass ActivatableClassId="App.PreInstallConfigTask" ThreadingModel="MTA"/>
      </InProcessServer>
    </Extension>
  </Extensions>

  <Applications>
    <Application>
      <Extensions>
        <Extension Category="windows.preInstalledConfigTask" EntryPoint=" App.PreInstallConfigTask">
        </Extension>
      </Extensions>
    </Application>
  </Applications>
</Package>

```

Here's the example C# code:

```

public sealed class PreInstallConfigTask : IBackgroundTask
{
    public async void Run(IBackgroundTaskInstance taskInstance)
    {
        CancellationTokenSource cts = new CancellationTokenSource();
        var deferral = taskInstance.GetDeferral();
        taskInstance.Canceled += 
            (sender, reason) =>
        {
            cts.Cancel();
        };
        try
        {
            await DownloadContactList(); // Do app migration/update steps.
        }
        catch (TaskCanceledException x)
        {
            // do nothing on cancellation.
        }
        deferral.Complete();
    }
}

```

## Preinstalls tasks for Classic Windows Apps

There are three deployments task types available to Classic Windows Apps, as shown in the table below. If you deploy your Classic Windows App on Windows 10, these tasks will work as expected.

TASK	DESCRIPTION
PREINSTALL_OEM_TASK	A 1st or 2nd party preinstalled app can run at install time task without requiring the app to be launched by the end user.
UPDATE_TASK	After an app has been updated, including .appx to .uap, a servicing task can be run to carry out any migration related tasks, also without requiring any user interaction.

TASK	DESCRIPTION
CONVERGENCE	Windows 8.1 and Windows 8.1 Phone code convergence. Also, enable unified .appx manifest schema.

# Change history for customization docs

10/2/2018 • 4 minutes to read • [Edit Online](#)

The following tables record the major changes that were made in the **Customize** section of the Windows 10 partner documentation since Windows 10, version 1607 was released.

## April 30, 2018

Changes in this section relate to the release of Windows 10, version 1803.

TOPIC	DESCRIPTION
<a href="#">Customize OOBE</a>	Updated with the new OOBE flow for Windows 10, version 1803. Added information about cloud service OOBE pages.
<a href="#">OOBE screen details</a>	Updated with details on two new OOBE screens that introduced in Windows 10, version 1803: the new payment information screen in the <b>Office Setup</b> portion of OOBE, and the <b>local account security questions</b> screen in the <b>Account setup</b> portion of OOBE.
<a href="#">Customize the Start layout</a>	Updated to reflect new customization options for the Microsoft suite of tiles in the Start layout, introduced in Windows 10, version 1803. Updated to reflect that apps no longer need to be pinned to the Start layout to remain installed on the device, as long as the <code>region</code> parameter in DISM is used when preinstalling the apps.
<a href="#">Customize SIM card slot names</a>	New. Describes how you can customize the names of SIM card slots on the device to more easily differentiate between them.
<a href="#">Shell Launcher</a>	Updated to reflect that in Windows 10, version 1803, you can configure Shell Launcher using the Assigned Access CSP.
<a href="#">Power controls</a>	Power controls include settings that control the system's power and behavior. In Windows 10, version 1803, two new settings have been added to Power controls: <a href="#">IgnoreCsComplianceCheck</a> , and <a href="#">EnableInputSuppression</a> .
<a href="#">Changed answer file settings for Windows 10, version 1803</a>	Learn about the Unattend settings that have been added, deprecated, and removed in the most recent version of Windows.

## January 2018

TOPIC	DESCRIPTION
<a href="#">OEM registration pages</a>	Updated. New screenshots and XML sample, clarifications on how the Oobe.xml elements relate to registration page fields, clarifications on collecting encrypted customer data.

## December 2017

TOPIC	DESCRIPTION
Windows updates during OOBE	New. Describes how both critical and non-critical Windows and driver updates are downloaded during a user's Out of Box Experience.
Exclusive apps	New. Guidance on how OEMs can work with software developers to target OEM devices for apps to appear exclusively on, based on the OEM IDs set in the registry.
Hibernate Once Resume Many	Updated to note that HORM (a feature of Unified Write Filter) can now be used on UEFI devices starting in Windows 10, version 1709.

## November 2017

TOPIC	DESCRIPTION
Customize OOBE	Updated with recommendation for setting the default volume level during OOBE.
Connect users to the network during OOBE	Updated with clarifications on how Cellular and Wi-Fi connections are used during OOBE, and the types of updates that download during OOBE.
Keyboard Filter	Updated to note that Keyboard Filter is not supported in a remote desktop session.
Unattend Setting: FirewallGroups	Updated with guidance on how to obtain the correct FirewallGroup-Group value using PowerShell.

## October 17, 2017

Changes in this section relate to the release of Windows 10, version 1709.

TOPIC	DESCRIPTION
Customize the Get Help app	New. Learn how to add your support app or website to Windows' self-service Get Help app, to provide customers with an easy-to-find way to reach out.
Customize the Windows performance power slider	New. The Windows performance power slider enables customers to trade performance of their system for longer battery life. You can configure the default slider mode, and the power settings engaged behind the scenes.
Customize a SAR mapping table	New. Configure and store a Specific Absorption Rate (SAR) table for mobile broadband modems in the registry.
Customize the Start layout	New. Customize the size of the start layout, and add your own tiles to it.

TOPIC	DESCRIPTION
Create a Kiosk Experience	Updated with guidance on providing a multi-app kiosk experience. This functionality is new in Windows 10 version 1709.
Adaptive hibernate	Updated. In Windows 10 version 1709, user usage prediction no longer triggers Hibernate. Also updated to include default values of hibernate triggers.
Predefined key combinations	Updated with keyboard shortcut changes introduced in Windows 10 version 1709.
OOBE.xml	Updated. In Windows 10 version 1709, <code>timezone</code> is now available to set in OOBE.xml
Changed answer file settings for Windows 10 version 1709	Learn about the Unattend settings that have been added, deprecated, and removed in the most recent version of Windows.

## September 27, 2017

TOPIC	DESCRIPTION
Customize the Out of Box Experience	New. Guidance on how to customize elements of the Out of Box Experience (OOBE), such as setting default values, adding registration screens, and providing support for unpaired mice and keyboards.

## March 24, 2017

NEW OR UPDATED TOPIC	DESCRIPTION
<a href="#">Microsoft-Windows-TPM-Tasks-ClearTpm</a>	New. Specifies whether to clear the Trusted Platform Module (TPM) during Windows setup. Clearing the TPM prevents an issue in earlier versions that kept some Windows features from working if the TPM was incorrectly set.
<a href="#">Microsoft-Windows-TwinUI-Hide</a>	New. Specifies whether to hide the link to an advanced settings app in the Pen and Windows Ink Settings page.
<a href="#">Preinstallable apps for Windows 10 Mobile</a>	Updated. Uses imggen.cmd to build the mobile image because ICD no longer includes support for image creation.

## March 15, 2017

NEW OR UPDATED TOPIC	DESCRIPTION

New or updated topic	Description
<a href="#">Customize the Country and Operator Settings Asset</a>	New. When a SIM is inserted in a COSA-enabled Windows-based device, the provisioning framework attempts to establish a cellular connection by searching for the matching profile and APN in COSA.

## October 6, 2016

New or updated topic	Description
<a href="#">Customize the taskbar</a>	New. Starting in Windows 10, version 1607, you can pin up to three additional apps to the taskbar by adding a taskbar layout modification file, for example, TaskbarLayoutModification.xml. You can specify different taskbar configurations based on SKU, device locale, or region.
<a href="#">Set dark mode</a>	New. Windows 10, build 1607 exposes a new personalization setting for end users, allowing them to express preference whether to see applications which support the setting in a dark or light mode.