Windows Embedded 8

Launch the system

Control user interaction

Build a custom OS

Windows Embedded 8.1 Industry provides branding and lockdown features to ensure that the industry devices you create reflect your corporate identity and help provide a targeted, secure, and consistent user experience.

Some of the branding and lockdown functionality described here requires the ELM Update.

Brand it and lock it down

Use custom branding features to customize the entire look and feel of your device, from the startup screen to the shutdown screen and everything in between.

Use lockdown features to deliver a targeted experience and ensure consistent configuration by limiting how users can interact with your device.

Add and configure settings

Add branding and lockdown features to your OS at design time by using either Control Panel or Deployment Image Servicing and Management (DISM). Some features are disabled by default and you must enable them after they are added to the OS. Assigned access is always part of the OS.

The methods for configuring specific settings for each feature at design time or run time vary from feature to feature.

Unbranded Boot

Use Unbranded Boot to suppress Windows 8.1 elements that appear when Industry 8.1 starts or resumes. You can also use Unbranded Boot to suppress the crash screen if Industry 8.1 encounters an error that it cannot recover from.

Design-time configuration
Configure Unbranded Boot settings at design time by using Windows System Image Manager (Windows SIM).

Run-time configuration
Configure Unbranded Boot settings at run time by using the BCDEdit command-line tool.

Custom Logon

Use Custom Logon to suppress Windows 8.1 UI elements related to the Welcome screen and shutdown screen. If used with autologon, Custom Logon will prevent the Welcome screen from appearing at all.

You can also use a custom credential provider to provide a custom sign-in experience for your device. You can use any credential provider that is compatible with Windows 8.1.

Design-time configuration
Configure Custom Logon settings at design time by using Windows SIM.

Run-time configuration
You cannot configure Custom Logon settings at run time.

Shell Launcher

Use Shell Launcher to replace the default Windows 8.1 shell with a custom shell. Use an application or executable as your custom shell, such as a command window or a custom dedicated application. Or launch different shell applications for different users or user groups.

However, to use a Windows Store app as a custom shell, use Windows 8 Application Launcher instead of Shell Launcher.

You can enable Shell Launcher at run time by using ELM or WMI providers.

Design-time configuration
Configure Shell Launcher settings at design time by using Windows SIM.

Run-time configuration
Configure Shell Launcher settings at run time by using Windows Management Instrumentation (WMI) providers directly in a Windows PowerShell script or in an application.

Or use Embedded Lockdown Manager (ELM) to directly configure a run-time image or to generate Windows PowerShell scripts that use the WMI providers.

Windows 8 Application Launcher

Use Windows 8 Application Launcher to start a Windows Store app immediately after a user signs in to a device. You can configure Windows 8 Application Launcher to start different apps for different user accounts.

For example, you might configure a device to launch one Windows Store app for customer accounts, but launch a different Windows Store app for employee accounts.

You can enable Windows 8 Application Launcher at run time by using ELM or WMI providers.

Design-time configuration
Configure Windows 8 Application Launcher settings at design time by using Windows SIM.

Run-time configuration
Configure Windows 8 Application Launcher settings at run time by using WMI providers directly in a Windows PowerShell script or in an application.

Or use ELM to directly configure a run-time image or to generate Windows PowerShell scripts that use the WMI providers. To configure this feature with ELM, you must have WEMSAL_UserAppXInformation installed.

Assigned access

Use assigned access to restrict a user account to access a single Windows Store application, which runs in full-screen mode.

For example, you can use assigned access to set up single-function devices, such as restaurant menus or displays at trade shows.

Assigned access is included in all Industry 8.1 operating systems.

Design-time configuration
You cannot configure assigned access at design time.

Run-time configuration
Configure assigned access settings at run time by accessing PC Settings or by using WMI providers directly in a Windows PowerShell script.

Gesture Filter

Use Gesture Filter to disable the new edge gestures available in Windows 8. Gesture Filter enables you to block top extended swipe and each of the edge gestures (left, right, and each corner) individually.

For example, you may want to prevent end users from swiping in from the right edge of the screen to access the charms.

You can use ELM or WMI providers to enable Gesture Filter at either design time or run time.

Design-time configuration
Configure Gesture Filter settings at design time by using Windows SIM.

Run-time configuration
Configure Gesture Filter settings at run time by using WMI providers directly in a Windows PowerShell script or in an application. Or use ELM to directly configure a run-time image or to generate Windows PowerShell scripts that use the WMI providers.

Keyboard Filter

Use Keyboard Filter to suppress undesirable keystrokes or key combinations. For example, you may not want customers to use Windows key combinations to lock the screen or use Task Manager.

Keyboard Filter works with physical keyboards, the touch keyboard, and the Windows On-Screen 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.

Design-time configuration
Configure Keyboard Filter settings at design time by using Windows SIM. However, you cannot set the breakout key at design time.

Run-time configuration
Configure Keyboard Filter settings at run time by using WMI providers directly in a Windows PowerShell script or in an application. Or use ELM to directly configure a run-time image or to generate Windows PowerShell scripts that use the WMI providers.

Dialog Filter

Use Dialog Filter to control which dialog boxes are displayed on the screen, and to automatically handle dialog boxes by taking a default action, such as to close or show the dialog box.

Or configure Dialog Filter to always show dialog boxes from specific processes, regardless of the specified default action.

Design-time configuration
Configure Dialog Filter settings at design time by using Windows SIM.

Run-time configuration
Configure Dialog Filter settings at run time by using WMI providers directly in a Windows PowerShell script or in an application. Or use ELM to directly configure a run-time image or to generate Windows PowerShell scripts that use the WMI providers.

Toast Notification Filter

Use Toast Notification Filter to prevent system toast notifications from displaying. It does not block toast notifications from applications.

You must use WMI providers to enable Toast Notification Filter at either design time or run time.

Design-time configuration
Configure Toast Notification settings at design time by using Windows SIM.

Run-time configuration
Configure Toast Notification Filter settings at run time by using WMI providers directly in a Windows PowerShell script.

USB Filter

Use USB Filter to block access to all USB ports except by trusted devices you have specifically permitted. You can identify trusted devices by device product ID, device vendor ID, or device class ID.

You can use ELM to enable USB Filter and allow all currently connected USB devices at run time.

Design-time configuration
Configure USB Filter settings at design time by using Windows SIM.

Run-time configuration
Use ELM to enable or disable selected USB ports for all devices. For more specific control, configure USB Filter settings at run time by using WMI providers directly in a Windows PowerShell script or in an application.

Unified Write Filter

Use Unified Write Filter (UWF) to provide support for stateless device operation, increase system reliability, and help reduce wear on hard drives.

UWF protects volumes by intercepting and redirecting writes to an overlay that records changes to protected volumes. Conceptually, an overlay is similar to a transparency on an overhead projector. Any change made to the transparency affects the projection while the underlying picture remains unchanged. You can protect a volume with UWF while excluding specific files, folders, or registry keys from being filtered.

You must enable UWF at run time by using WMI providers.

Design-time configuration
Configure UWF settings at design time by using Windows SIM.

Run-time configuration
Configure UWF settings at run time by using WMI providers directly in a Windows PowerShell script or in an application. Or use ELM to directly configure a run-time image or to generate Windows PowerShell scripts that use the WMI providers. Or use the command-line tool UWFMgr.exe.

Lockdown and branding features for Windows Embedded 8.1 Industry

Microsoft

Feedback on this diagram? Send email to webfdbk@microsoft.com.