Table of Contents

Microsoft Surface Hub

5/4/2017 • 1 min to read • Edit Online

Documents related to deploying and managing the Microsoft Surface Hub in your organization.

> Looking for the user's guide for Surface Hub?

In this section

TOPIC	DESCRIPTION
Microsoft Surface Hub administrator's guide	This guide covers the installation and administration of devices running Surface Hub, and is intended for use by anyone responsible for these tasks, including IT administrators and developers.
Differences between Surface Hub and Windows 10 Enterprise	This topic explains the differences between the operating system on Surface Hub and Windows 10 Enterprise.
How Surface Hub addresses Wi-Fi Direct security issues	This topic provides guidance on Wi-Fi Direct security risks, how the Surface Hub has addressed those risks, and how Surface Hub administrators can configure the device for the highest level of security.
Useful downloads for Surface Hub administrators	This topic provides links to useful Surface Hub documents, such as product datasheets, the site readiness guide, and user's guide.
Change history for Surface Hub	This topic lists new and updated topics in the Surface Hub documentation.

Microsoft Surface Hub administrator's guide

5/4/2017 • 1 min to read • Edit Online

This guide covers the installation and administration of devices running Surface Hub, and is intended for use by anyone responsible for these tasks, including IT administrators and developers.

Before you power on Microsoft Surface Hub for the first time, make sure you've completed preparation items, and that you have the information listed in the Setup worksheet. When you do power it on, the device will walk you through a series of setup screens. If you haven't properly set up your environment, or don't have the required information, you'll have to do extra work afterward making sure the settings are correct.

In this section

TOPIC	DESCRIPTION
Intro to Microsoft Surface Hub	Surface Hub is an all-in-one productivity device that is intended for brainstorming, collaboration, and presentations. In order to get the maximum benefit from Surface Hub, your organization's infrastructure and the Surface Hub itself must be properly set up and integrated. This guide describes what needs to be done both before and during setup in order to help you optimize your use of the device.
Physically install Microsoft Surface Hub	The Surface Hub Readiness Guide will help make sure that your site is ready for the installation. You can download the Guide from the Microsoft Download Center. It includes planning information for both the 55" and 84" devices, as well as info on moving the Surface Hub from receiving to the installation location, mounting options, and a list of what's in the box.
Prepare your environment for Microsoft Surface Hub	This section contains an overview of the steps required to prepare your environment so that you can use all of the features of Surface Hub. See Intro to Surface Hub for a description of how the device and its features interact with your IT environment.
Set up Microsoft Surface Hub	Set up instructions for Surface Hub include a setup worksheet, and a walkthrough of the first-run program.
Manage Microsoft Surface Hub	How to manage your Surface Hub after finishing the first-run program.
Troubleshoot Microsoft Surface Hub	Troubleshoot common problems, including setup issues, Exchange ActiveSync errors.

TOPIC	DESCRIPTION
Appendix: PowerShell	PowerShell scripts to help set up and manage your Surface Hub .

Intro to Microsoft Surface Hub

5/4/2017 • 1 min to read • Edit Online

Microsoft Surface Hub is an all-in-one productivity device that is intended for brainstorming, collaboration, and presentations. In order to get the maximum benefit from Surface Hub, your organization's infrastructure and the Surface Hub itself must be properly set up and integrated. This guide describes what needs to be done both before and during setup in order to help you optimize your use of the device.

You'll need to understand how each of these services interacts with Surface Hub. See Prepare your environment for Surface Hub for details.

Surface Hub setup process

In some ways, adding your new Surface Hub is just like adding any other Microsoft Windows-based device to your network. However, in order to get your Surface Hub up and running at its full capacity, there are some very specific requirements. Here are the next topics you'll need:

1. Prepare your environment for Surface Hub
2. Physically install your Surface Hub device
3. Run the Surface Hub first-run setup program (OOBE)

Prepare your environment for Microsoft Surface Hub

5/4/2017 • 4 min to read • Edit Online

This section contains an overview of setup dependencies and the setup process. Review the info in this section to help you prepare your environment and gather information needed to set up your Surface Hub.

Review infrastructure dependencies

Review these dependencies to make sure Surface Hub features will work in your IT infrastructure.

DEPENDENCY	PURPOSE
Active Directory or Azure Active Directory (Azure AD)	The Surface Hub's uses an Active Directory or Azure AD account (called a **device account**) to access Exchange and Skype for Business services. The Surface Hub must be able to connect to your Active Directory domain controller or to your Azure AD tenant in order to validate the device account's credentials, as well as to access information like the device account's display name, alias, Exchange server, and Session Initiation Protocol (SIP) address. You can also domain join or Azure AD join your Surface Hub to allow a group of authorized users to configure settings on the Surface Hub.
Exchange (Exchange 2013 or later, or Exchange Online) and Exchange ActiveSync	Exchange is used for enabling mail and calendar features, and also lets people who use the device send meeting requests to the Surface Hub, enabling one-touch meeting join. ActiveSync is used to sync the device account's calendar and mail to the Surface Hub. If the device cannot use ActiveSync, it will not show meetings on the welcome screen, and joining meetings and emailing whiteboards will not be enabled.
Skype for Business (Lync Server 2013 or later, or Skype for Business Online)	Skype for Business is used for various conferencing features, like video calls, instant messaging, and screen sharing. If screen sharing on a Surface Hub fails and the error message **An error occurred during the screen presentation** is displayed, see Video Based Screen Sharing not working on Surface Hub for help.
Mobile device management (MDM) solution (Microsoft Intune, System Center Configuration Manager, or supported third-party MDM provider)	If you want to apply settings and install apps remotely, and to multiple devices at a time, you must set up a MDM solution and enroll the device to that solution. See Manage settings with an MDM provider for details.
Microsoft Operations Managmement Suite (OMS)	OMS is used to monitor the health of Surface Hub devices. See Monitor your Surface Hub for details.

DEPENDENCY	PURPOSE
Network and Internet access	In order to function properly, the Surface Hub should have access to a wired or wireless network. Overall, a wired connection is preferred. **Dynamic IP:** The Surface Hub cannot be configured to use a static IP. It must use DHCP to assign an IP address. **Proxy servers:** If your topology requires a connection to a proxy server to reach Internet services, then you can configure it during first run, or in Settings.

Additionally, note that Surface Hub requires the following open ports:

- HTTPS: 443
- HTTP: 80

Depending on your environment, access to additional ports may be needed:

- For online environments, see Office 365 IP URLs and IP address ranges.
- For on-premises installations, see Skype for Business Server: Ports and protocols for internal servers.

Microsoft collects telemetry to help improve your Surface Hub experience. Add these sites to your allow list:

- Telemetry client endpoint: `https://vortex.data.microsoft.com/`
- Telemetry settings endpoint: `https://settings.data.microsoft.com/`

Work with other admins

Surface Hub interacts with a few different products and services. Depending on the size of your organization, there could be multiple people supporting different products in your environment. You'll want to include people who manage Exchange, Active Directory (or Azure Active Directory), mobile device management (MDM), and network resources in your planning and prep for Surface Hub deployments.

Create and verify device account

A device account is an Exchange resource account that Surface Hub uses to display its meeting calendar, join Skype for Business calls, and send email. See Create and test a device account for details.

After you've created your device account, there are a couple of ways to verify that it's setup correctly.

- Run Surface Hub device account validation PowerShell scripts. For more information, see Surface Hub device account scripts in Script Center, or PowerShell scripts for Surface Hub later in this guide.
- Use the account with the Lync Microsoft Store app. If Lync signs in successfully, then the device account will most likely work with Skype for Business on Surface Hub.

Prepare for first-run program

There are a few more item to consider before you start the first-run program.

Create provisioning packages (optional)

You can use provisioning packages to add certificates, customize settings and install apps. See Create provisioning packages for details. You can install provisioning packages at first-run.

Set up admin groups

Every Surface Hub can be configured locally using the Settings app on the device. To prevent unauthorized users

from changing settings, the Settings app requires admin credentials to open the app. See Admin group management for details on how admin groups are set up and managed. You will set up admins for the device at first run.

Review and complete Surface Hub setup worksheet (optional)

When you go through the first-run program for your Surface Hub, there's some information that you'll need to supply. The setup worksheet summarizes that info, and provides lists of environment-specific info that you'll need when you go through the first-run program. For more information, see Setup worksheet.

In this section

TOPIC	DESCRIPTION
Create and test a device account	This topic introduces how to create and test the device account that Surface Hub uses to communicate with and Skype.
Create provisioning packages	For Windows 10, settings that use the registry or a content services platform (CSP) can be configured using provisioning packages. You can also add certificates during first run using provisioning.
Admin group management	Every Surface Hub can be configured individually by opening the Settings app on the device. However, to prevent people who are not administrators from changing the settings, the Settings app requires administrator credentials to open the app and change settings. The Settings app requires local administrator credentials to open the app.

Physically install Microsoft Surface Hub

5/4/2017 • 1 min to read • Edit Online

The Microsoft Surface Hub Readiness Guide will help make sure that your site is ready for the installation. You can download the Guide from the Microsoft Download Center. It includes planning information for both the 55" and 84" devices, as well as info on moving the Surface Hub from receiving to the installation location, mounting options, and a list of what's in the box.

You may also want to check out the Unpacking Guide. It will show you how to unpack the devices efficiently and safely. There are two guides, one for the 55" and one for the 84". A printed version of the Unpacking Guide is attached to the outside front of each unit's shipping crate.

- Download the 55" Unpacking Guide from the Microsoft Download Center.
- Download the 84" version from the Microsoft Download Center.

Create and test a device account (Surface Hub)

5/4/2017 • 2 min to read • Edit Online

This topic introduces how to create and test the device account that Microsoft Surface Hub uses to communicate with Microsoft Exchange and Skype.

A **device account** is an Exchange resource account that Surface Hub uses to:

- Display its meeting calendar
- Join Skype for Business calls
- Send email (for example, email whiteboard content from a meeting)

Once the device account is provisioned to a Surface Hub, people can add this account to a meeting invitation the same way that they would invite a meeting room.

Configuration overview

This table explains the main steps and configuration decisions when you create a device account.

STEP	DESCRIPTION	PURPOSE
1	Created a logon-enabled Exchange resource mailbox (Exchange 2013 or later, or Exchange Online)	This resource mailbox allows the device to maintain a meeting calendar, receive meeting requests, and send mail. It must be logon-enabled to be provisioned to a Surface Hub.
2	Configure mailbox properties	The mailbox must be configured with the correct properties to enable the best meeting experience on Surface Hub. For more information on mailbox properties, see Mailbox properties.
3	Apply a compatible mobile device mailbox policy to the mailbox	Surface Hub is managed using mobile device management (MDM) rather than through mobile device mailbox policies. For compatibility, the device account must have a mobile device mailbox policy where the **PasswordEnabled** setting is set to False. Otherwise, Surface Hub can't sync mail and calendar info.
4	Enable mailbox with Skype for Business (Lync Server 2013 or later, or Skype for Business Online)	Skype for Business must be enabled to use conferencing features like video calls, IM, and screen sharing.
5	(Optional) Whitelist ActiveSync Device ID	Your organization may have a global policy that prevents device accounts from syncing mail and calendar info. If so, you need to whitelist the ActiveSync Device ID of your Surface Hub.

STEP	DESCRIPTION	PURPOSE
6	(Optional) Disable password expiration	To simplify management, you can turn off password expiration for the device account and allow Surface Hub to automatically rotate the device account password. For more information about password management, see Password management.

Detailed configuration steps

We recommend setting up your device accounts using remote PowerShell. There are PowerShell scripts available to help create and validate device accounts For more information on PowerShell scripts and instructions, see Appendix A: PowerShell.

For detailed steps using PowerShell to provision a device account, choose an option from the table, based on your organization deployment.

ORGANIZATION DEPLOYMENT	DESCRIPTION
Online deployment (Office 365)	Your organization's environment is deployed entirely on Office 365.
On-premises deployment (single-forest)	Your organization has servers that it controls and uses to host Active Directory, Exchange, and Skype for Business (or Lync) in a single-forest environment.
On-premises deployment (multiple forests)	Your organization has servers that it controls and uses to host Active Directory, Exchange, and Skype for Business (or Lync) in a multi-forest environment.
Hybrid deployment	Your organization has a mix of services, with some hosted on-premises and some hosted online through Office 365.
Online or hybrid deployment using Skype Hybrid Voice environment	Your organization has Skype for Business home pools and Exchange servers in the cloud, and uses an on-premises pool of Skype for Business 2015 or Cloud Connector edition connected via Public Switched Telephone Network (PSTN).

If you prefer to use a graphical user interface (UI), some steps can be done using UI instead of PowerShell. For more information, see Creating a device account using UI.

Online deployment with Office 365 (Surface Hub)

5/4/2017 • 3 min to read • Edit Online

This topic has instructions for adding a device account for your Microsoft Surface Hub when you have a pure, online deployment.

If you have a pure, online (O365) deployment, then you can use the provided PowerShell scripts to create device accounts.

1. Start a remote PowerShell session on a PC and connect to Exchange.

   Be sure you have the right permissions set to run the associated cmdlets.

   ```
   Set-ExecutionPolicy Unrestricted
   $org='contoso.microsoft.com'
   $cred=Get-Credential admin@$org
   $sess= New-PSSession -ConfigurationName Microsoft.Exchange -ConnectionUri
   https://outlook.office365.com/powershell-liveid/ -Credential $cred -Authentication Basic -
   AllowRedirection
   Import-PSSession $sess
   ```

2. After establishing a session, you'll either create a new mailbox and enable it as a RoomMailboxAccount, or change the settings for an existing room mailbox. This will allow the account to authenticate into the Surface Hub.

   If you're changing an existing resource mailbox:

   ```
   Set-Mailbox -Identity 'HUB01' -EnableRoomMailboxAccount $true -RoomMailboxPassword (ConvertTo-
   SecureString -String <password> -AsPlainText -Force)
   ```

   If you're creating a new resource mailbox:

   ```
   New-Mailbox -MicrosoftOnlineServicesID HUB01@contoso.com -Alias HUB01 -Name "Hub-01" -Room -
   EnableRoomMailboxAccount $true -RoomMailboxPassword (ConvertTo-SecureString -String <password> -
   AsPlainText -Force)
   ```

3. After setting up the mailbox, you will need to either create a new Exchange ActiveSync policy, or use a compatible existing policy.

   Surface Hubs are only compatible with device accounts that have an ActiveSync policy where the **PasswordEnabled** property is set to False. If this isn't set properly, then Exchange services on the Surface Hub (mail, calendar, and joining meetings), will not be enabled.

   If you haven't created a compatible policy yet, use the following cmdlet—this one creates a policy called "Surface Hubs". Once it's created, you can apply the same policy to other device accounts.

   ```
   $easPolicy = New-MobileDeviceMailboxPolicy -Name "SurfaceHubs" -PasswordEnabled $false -
   AllowNonProvisionableDevices $True
   ```

   Once you have a compatible policy, then you will need to apply the policy to the device account.

```
Set-CASMailbox 'HUB01@contoso.com' -ActiveSyncMailboxPolicy $easPolicy.Id
```

4. Various Exchange properties must be set on the device account to improve the meeting experience. You can see which properties need to be set in the Exchange properties section.

```
Set-CalendarProcessing -Identity 'HUB01@contoso.com' -AutomateProcessing AutoAccept -
AddOrganizerToSubject $false –AllowConflicts $false –DeleteComments $false -DeleteSubject $false -
RemovePrivateProperty $false
Set-CalendarProcessing -Identity 'HUB01@contoso.com' -AddAdditionalResponse $true -AdditionalResponse
"This is a Surface Hub room!"
```

5. Connect to Azure AD.

   You need to connect to Azure AD to apply some account settings. You can run this cmdlet to connect.

```
Connect-MsolService -Credential $cred
```

6. If you decide to have the password not expire, you can set that with PowerShell cmdlets too. See Password management for more information.

```
Set-MsolUser -UserPrincipalName 'HUB01@contoso.com' -PasswordNeverExpires $true
```

7. Surface Hub requires a license for Skype for Business functionality.

   - Your Surface Hub account requires a Lync Online (Plan 2) or Lync Online (Plan 3) license, but it does not require an Exchange Online license.
   - You'll need to have Lync Online (Plan 2) or higher in your O365 plan. The plan needs to support conferencing capability.
   - If you need Enterprise Voice (PSTN telephony) using telephony service providers for the Surface Hub, you need Lync Online (Plan 3).

   Next, you can use `Get-MsolAccountSku` to retrieve a list of available SKUs for your O365 tenant.

   Once you list out the SKUs, you can add a license using the `Set-MsolUserLicense` cmdlet. In this case, `$strLicense` is the SKU code that you see (for example, *contoso:STANDARDPACK*).

```
Set-MsolUser -UserPrincipalName 'HUB01@contoso.com' -UsageLocation "US"
Get-MsolAccountSku
Set-MsolUserLicense -UserPrincipalName 'HUB01@contoso.com' -AddLicenses $strLicense
```

8. Enable the device account with Skype for Business.

   - Start by creating a remote PowerShell session from a PC.

```
Import-Module LyncOnlineConnector
$cssess=New-CsOnlineSession -Credential $cred
Import-PSSession $cssess -AllowClobber
```

   - Next, if you aren't sure what value to use for the `RegistrarPool` parameter in your environment, you can get the value from an existing Skype for Business user using this cmdlet (for example, *alice@contoso.com*):

```
Get-CsOnlineUser -Identity ‘alice@contoso.com’| fl *registrarpool*
```

OR by setting a variable

```
$strRegistrarPool = (Get-CsOnlineUser -Identity ‘alice@contoso.com’).RegistrarPool
```

- Enable the Surface Hub account with the following cmdlet:

```
Enable-CsMeetingRoom -Identity 'HUB01@contoso.com' -RegistrarPool yourRegistrarPool -
SipAddressType EmailAddress
```

OR using the $strRegistarPool variable from above

```
Enable-CsMeetingRoom -Identity 'HUB01@contoso.com' -RegistrarPool $strRegistrarPool -
SipAddressType EmailAddress
```

For validation, you should be able to use any Skype for Business client (PC, Android, etc) to sign in to this account.

On-premises deployment for Surface Hub in a single-forest environment

5/4/2017 • 3 min to read • Edit Online

This topic explains how you add a device account for your Microsoft Surface Hub when you have a single-forest, on-premises deployment.

If you have a single-forest on-premises deployment with Microsoft Exchange 2013 or later and Skype for Business 2013 or later, then you can use the provided PowerShell scripts to create device accounts. If you're using a multi-forest deployment, see On-premises deployment for Surface Hub in a multi-forest environment.

1. Start a remote PowerShell session from a PC and connect to Exchange.

   Be sure you have the right permissions set to run the associated cmdlets.

   Note here that `$strExchangeServer` is the fully qualified domain name (FQDN) of your Exchange server, and `$strLyncFQDN` is the FQDN of your Skype for Business server.

   ```
   Set-ExecutionPolicy Unrestricted
   $org='contoso.microsoft.com'
   $cred=Get-Credential $admin@$org
   $sessExchange = New-PSSession -ConfigurationName microsoft.exchange -Credential $cred -AllowRedirection
   -Authentication Kerberos -ConnectionUri "http://$strExchangeServer/powershell" -WarningAction
   SilentlyContinue
   $sessLync = New-PSSession -Credential $cred -ConnectionURI "https://$strLyncFQDN/OcsPowershell" -
   AllowRedirection -WarningAction SilentlyContinue
   Import-PSSession $sessExchange
   Import-PSSession $sessLync
   ```

2. After establishing a session, you'll either create a new mailbox and enable it as a RoomMailboxAccount, or change the settings for an existing room mailbox. This will allow the account to authenticate into the Surface Hub.

   If you're changing an existing resource mailbox:

   ```
   Set-Mailbox -Identity 'HUB01' -EnableRoomMailboxAccount $true -RoomMailboxPassword (ConvertTo-
   SecureString -String <password> -AsPlainText -Force)
   ```

   If you're creating a new resource mailbox:

   ```
   New-Mailbox -UserPrincipalName HUB01@contoso.com -Alias HUB01 -Name "Hub-01" -Room -
   EnableRoomMailboxAccount $true -RoomMailboxPassword (ConvertTo-SecureString -String <password> -
   AsPlainText -Force)
   ```

3. After setting up the mailbox, you will need to either create a new Exchange ActiveSync policy, or use a compatible existing policy.

   Surface Hubs are only compatible with device accounts that have an ActiveSync policy where the **PasswordEnabled** property is set to False. If this isn't set properly, then Exchange services on the Surface Hub (mail, calendar, and joining meetings), will not be enabled.

   If you haven't created a compatible policy yet, use the following cmdlet—this one creates a policy called

"Surface Hubs". Once it's created, you can apply the same policy to other device accounts.

```
$easPolicy = New-MobileDeviceMailboxPolicy -Name “SurfaceHubs” -PasswordEnabled $false
```

Once you have a compatible policy, then you will need to apply the policy to the device account. However, policies can only be applied to user accounts and not resource mailboxes. You need to convert the mailbox into a user type, apply the policy, and then convert it back into a mailbox—you may need to re-enable it and set the password again too.

```
Set-Mailbox $acctUpn -Type Regular
Set-CASMailbox $acctUpn -ActiveSyncMailboxPolicy $easPolicy
Set-Mailbox $acctUpn -Type Room
Set-Mailbox $credNewAccount.UserName -RoomMailboxPassword $credNewAccount.Password -
EnableRoomMailboxAccount $true
```

4. Various Exchange properties can be set on the device account to improve the meeting experience for people. You can see which properties need to be set in the Exchange properties section.

```
Set-CalendarProcessing -Identity $acctUpn -AutomateProcessing AutoAccept -AddOrganizerToSubject $false –
AllowConflicts $false –DeleteComments $false -DeleteSubject $false -RemovePrivateProperty $false
Set-CalendarProcessing -Identity $acctUpn -AddAdditionalResponse $true -AdditionalResponse "This is a
Surface Hub room!"
```

5. If you decide to have the password not expire, you can set that with PowerShell cmdlets too. See Password management for more information.

```
Set-AdUser $acctUpn -PasswordNeverExpires $true
```

6. Enable the account in Active Directory so it will authenticate to the Surface Hub.

```
Set-AdUser $acctUpn -Enabled $true
```

7. Enable the device account with Skype for Business by enabling your Surface Hub AD account on a Skype for Business Server pool:

```
Enable-CsMeetingRoom -SipAddress "sip:HUB01@contoso.com"
 -DomainController DC-ND-001.contoso.com -RegistrarPool LYNCPool15.contoso.com
 -Identity HUB01
```

You'll need to use the Session Initiation Protocol (SIP) address and domain controller for the Surface Hub, along with your own Skype for Business Server pool identifier and user identity.

8. OPTIONAL: You can also allow your Surface Hub to make and receive public switched telephone network (PSTN) phone calls by enabling Enterprise Voice for your account. Enterprise Voice isn't a requirement for Surface Hub, but if you want PSTN dialing functionality for the Surface Hub client, here's how to enable it:

```
Set-CsMeetingRoom HUB01 -DomainController DC-ND-001.contoso.com
 -LineURItel: +14255550555;ext=50555" Set-CsMeetingRoom -DomainController DC-ND-001.contoso.com
 -Identity HUB01 -EnterpriseVoiceEnabled $true
```

Again, you'll need to replace the provided domain controller and phone number examples with your own information. The parameter value `$true` stays the same.

On-premises deployment for Surface Hub in a multi-forest environment

5/4/2017 • 2 min to read • Edit Online

This topic explains how you add a device account for your Microsoft Surface Hub when you have a multi-forest, on-premises deployment.

If you have a multi-forest on-premises deployment with Microsoft Exchange 2013 or later and Skype for Business 2013 or later, then you can use the provided PowerShell scripts to create device accounts. If you're using a single-forest deployment, see On-premises deployment for Surface Hub in a single-forest environment.

1. Start a remote PowerShell session from a PC and connect to Exchange.

   Be sure you have the right permissions set to run the associated cmdlets.

   Note here that `$strExchangeServer` is the fully qualified domain name (FQDN) of your Exchange server, and `$strLyncFQDN` is the FQDN of your Skype for Business server.

   ```
   Set-ExecutionPolicy Unrestricted
   $org='contoso.microsoft.com'
   $cred=Get-Credential $admin@$org
   $sessExchange = New-PSSession -ConfigurationName microsoft.exchange -Credential $cred -AllowRedirection
   -Authentication Kerberos -ConnectionUri "http://$strExchangeServer/powershell" -WarningAction
   SilentlyContinue
   $sessLync = New-PSSession -Credential $cred -ConnectionURI "https://$strLyncFQDN/OcsPowershell" -
   AllowRedirection -WarningAction SilentlyContinue
   Import-PSSession $sessExchange
   Import-PSSession $sessLync
   ```

2. After establishing a session, create a new mailbox in the Resource Forest. This will allow the account to authenticate into the Surface Hub.

   If you're changing an existing resource mailbox:

   ```
   New-Mailbox -UserPrincipalName HUB01@contoso.com -Alias HUB01 -Name "Hub-01"
   ```

3. After setting up the mailbox, you will need to either create a new Exchange ActiveSync policy, or use a compatible existing policy.

   Surface Hubs are only compatible with device accounts that have an ActiveSync policy where the **PasswordEnabled** property is set to **False**. If this isn't set properly, then Exchange services on the Surface Hub (mail, calendar, and joining meetings), will not be enabled.

   If you haven't created a compatible policy yet, use the following cmdlet-—this one creates a policy called "Surface Hubs". Once it's created, you can apply the same policy to other device accounts.

   ```
   $easPolicy = New-MobileDeviceMailboxPolicy -Name “SurfaceHubs” -PasswordEnabled $false
   ```

   Once you have a compatible policy, then you will need to apply the policy to the device account.

```
Set-CASMailbox $acctUpn -ActiveSyncMailboxPolicy $easPolicy -ActiveSyncEnabled $true
Set-Mailbox $acctUpn -Type Room
```

4. Various Exchange properties can be set on the device account to improve the meeting experience for people. You can see which properties need to be set in the Exchange properties section.

```
Set-CalendarProcessing -Identity $acctUpn -AutomateProcessing AutoAccept -AddOrganizerToSubject $false –
AllowConflicts $false –DeleteComments $false -DeleteSubject $false -RemovePrivateProperty $false
Set-CalendarProcessing -Identity $acctUpn -AddAdditionalResponse $true -AdditionalResponse "This is a
Surface Hub room!"
```

5. If you decide to have the password not expire, you can set that with PowerShell cmdlets too. See Password management for more information. This should be set in the User Forest.

```
Set-AdUser $acctUpn -PasswordNeverExpires $true
```

6. Enable the account in Active Directory so it will authenticate to the Surface Hub. This should be set in the User Forest.

```
Set-AdUser $acctUpn -Enabled $true
```

7. You now need to change the room mailbox to a linked mailbox:

```
$cred=Get-Credential AuthForest\LinkedRoomTest1
Set-mailbox -Alias LinkedRoomTest1 -LinkedMasterAccount AuthForest\LinkedRoomTest1 -
LinkedDomainController AuthForest-4939.AuthForest.extest.contoso.com -Name LinkedRoomTest1 -
LinkedCredential $cred -Identity LinkedRoomTest1
```

8. Enable the device account with Skype for Business by enabling your Surface Hub AD account on a Skype for Business Server pool:

```
Enable-CsMeetingRoom -SipAddress "sip:HUB01@contoso.com"
 -DomainController DC-ND-001.contoso.com -RegistrarPool LYNCPool15.contoso.com
 -Identity HUB01
```

   You'll need to use the Session Initiation Protocol (SIP) address and domain controller for the Surface Hub, along with your own Skype for Business Server pool identifier and user identity.

Hybrid deployment (Surface Hub)

5/4/2017 • 15 min to read • Edit Online

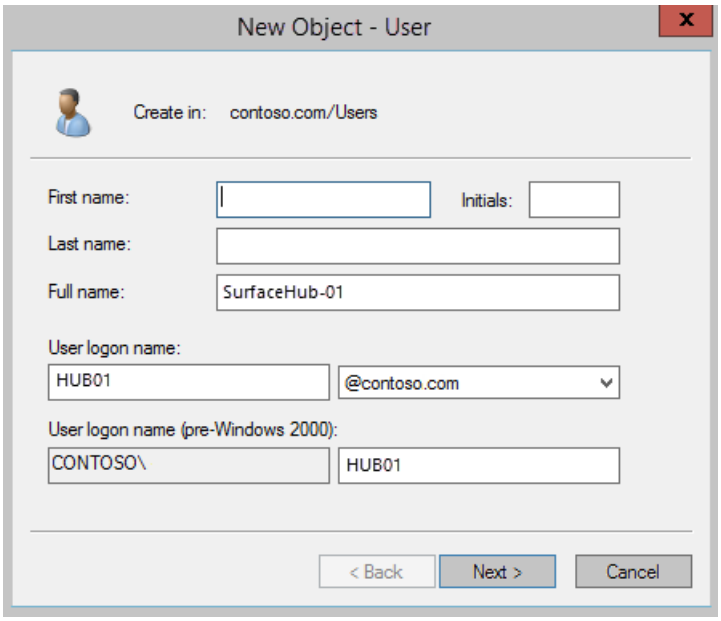
A hybrid deployment requires special processing to set up a device account for your Microsoft Surface Hub. If you're using a hybrid deployment, in which your organization has a mix of services, with some hosted on-premises and some hosted online, then your configuration will depend on where each service is hosted. This topic covers hybrid deployments for Exchange hosted on-prem, Exchange hosted online, Skype for Business on-prem, Skype for Business online, and Skype for Business hybrid. Because there are so many different variations in this type of deployment, it's not possible to provide detailed instructions for all of them. The following process will work for many configurations. If the process isn't right for your setup, we recommend that you use PowerShell (see Appendix: PowerShell) to achieve the same end result as documented here, and for other deployment options. You should then use the provided Powershell script to verify your Surface Hub setup. (See Account Verification Script.)

Exchange on-prem

Use this procedure if you use Exchange on-prem.

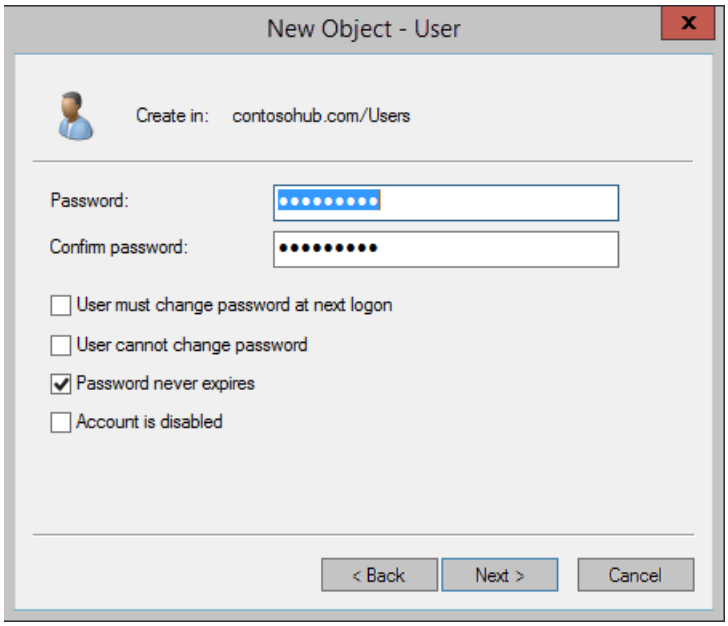
1. For this procedure, you'll be using AD admin tools to add an email address for your on-prem domain account. This account will be synced to Office 365.

   - In **Active Directory Users and Computers** AD tool, right-click on the folder or Organizational Unit that your Surface Hub accounts will be created in, click **New**, and **User**.
   - Type the display name from the previous cmdlet into the **Full name** box, and the alias into the **User logon name** box. Click **Next**.

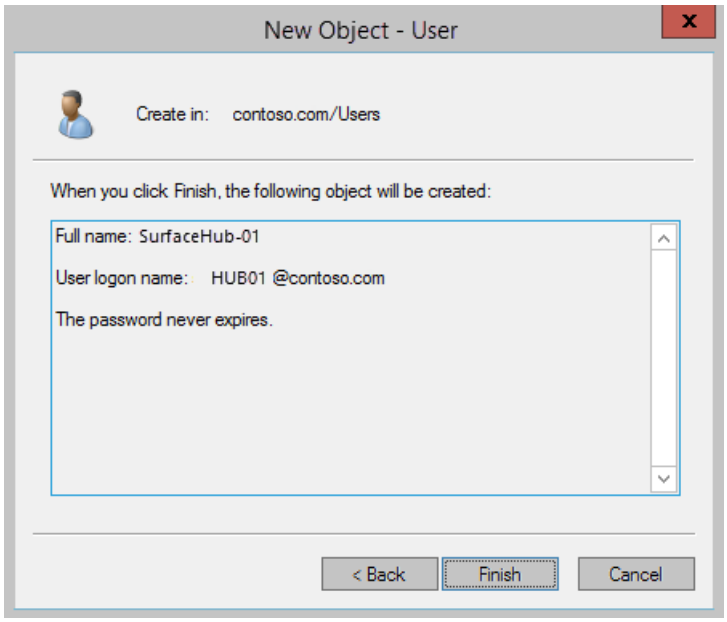


   - Type the password for this account. You'll need to retype it for verification. Make sure the **Password never expires** checkbox is the only option selected.

> **Important** Selecting **Password never expires** is a requirement for Skype for Business on the Surface Hub. Your domain rules may prohibit passwords that don't expire. If so, you'll need to create an exception for each Surface Hub device account.



- Click **Finish** to create the account.



2. After you've created the account, run a directory synchronization. When it's complete, go to the users page in your Office 365 admin center and verify that the account created in the previous steps has merged to online.

3. Enable the remote mailbox.

   Open your on-prem Exchange Management Shell with administrator permissions, and run this cmdlet.

   ```
   Enable-RemoteMailbox 'HUB01@contoso.com' -RemoteRoutingAddress 'HUB01@contoso.com' -Room
   ```

4. Connect to Microsoft Exchange Online and set some properties for the account in Office 365.

   Start a remote PowerShell session on a PC and connect to Microsoft Exchange. Be sure you have the right permissions set to run the associated cmdlets.

   The next steps will be run on your Office 365 tenant.

```
Set-ExecutionPolicy Unrestricted
$cred=Get-Credential -Message "Please use your Office 365 admin credentials"
$sess= New-PSSession -ConfigurationName Microsoft.Exchange -ConnectionUri
'https://ps.outlook.com/powershell' -Credential $cred -Authentication Basic -AllowRedirection
Import-PSSession $sess
```

5. Create a new Exchange ActiveSync policy, or use a compatible existing policy.

   After setting up the mailbox, you will need to either create a new Exchange ActiveSync policy or use a compatible existing policy.

   Surface Hubs are only compatible with device accounts that have an ActiveSync policy where the **PasswordEnabled** property is set to False. If this isn't set properly, then Exchange services on the Surface Hub (mail, calendar, and joining meetings), will not be enabled.

   If you haven't created a compatible policy yet, use the following cmdlet—-this one creates a policy called "Surface Hubs". Once it's created, you can apply the same policy to other device accounts.

   ```
   $easPolicy = New-MobileDeviceMailboxPolicy -Name "SurfaceHubs" -PasswordEnabled $false
   ```

   Once you have a compatible policy, then you will need to apply the policy to the device account. However, policies can only be applied to user accounts and not to resource mailboxes. You'll need to convert the mailbox into a user type, apply the policy, and then convert it back into a mailbox; you may need to re-enable it and set the password again too.

   ```
   Set-Mailbox 'HUB01@contoso.com' -Type Regular
   Set-CASMailbox 'HUB01@contoso.com' -ActiveSyncMailboxPolicy $easPolicy.id
   Set-Mailbox 'HUB01@contoso.com' -Type Room
   $credNewAccount = Get-Credential -Message "Please provide the Surface Hub username and password"
   Set-Mailbox 'HUB01@contoso.com' -RoomMailboxPassword $credNewAccount.Password -EnableRoomMailboxAccount
   $true
   ```

6. Set Exchange properties.

   Setting Exchange properties on the device account to improve the meeting experience. You can see which properties need to be set in the Exchange properties section.

   ```
   Set-CalendarProcessing -Identity 'HUB01@contoso.com' -AutomateProcessing AutoAccept -
   AddOrganizerToSubject $false –AllowConflicts $false –DeleteComments $false -DeleteSubject $false -
   RemovePrivateProperty $false
   Set-CalendarProcessing -Identity 'HUB01@contoso.com' -AddAdditionalResponse $true -AdditionalResponse
   'This is a Surface Hub room!'
   ```

7. Connect to Azure AD.

   You need to connect to Azure AD to apply some account settings. You can run this cmdlet to connect.

   ```
   Connect-MsolService -Credential $cred
   ```

8. Assign an Office 365 license.

   The device account needs to have a valid Office 365 (O365) license, or Exchange and Skype for Business will not work. If you have the license, you need to assign a usage location to your device account—this determines what license SKUs are available for your account.

   Next, you can use `Get-MsolAccountSku` to retrieve a list of available SKUs for your O365 tenant.

Once you list out the SKUs, you can add a license using the `Set-MsolUserLicense` cmdlet. In this case, `$strLicense` is the SKU code that you see (for example, *contoso:STANDARDPACK*).

```
Set-MsolUser -UserPrincipalName 'HUB01@contoso.com' -UsageLocation 'US'
Get-MsolAccountSku
Set-MsolUserLicense -UserPrincipalName 'HUB01@contoso.com' -AddLicenses $strLicense
```

Next, you enable the device account with Skype for Business Online, Skype for Business on-prem, or Skype for Business hybrid.

Skype for Business Online

To enable Skype for Business online, your environment will need to meet the following prerequisites:

- You need to have Lync Online (Plan 2) or higher in your O365 plan. The plan needs to support conferencing capability.
- If you need Enterprise Voice (PSTN telephony) using telephony service providers for the Surface Hub, you need Lync Online (Plan 3).
- Your tenant users must have Exchange mailboxes (at least one Exchange mailbox in the tenant is required).
- Your Surface Hub account does require a Lync Online (Plan 2) or Lync Online (Plan 3) license, but it does not require an Exchange Online license.

1. Start by creating a remote PowerShell session from a PC to the Skype for Business online environment.

   ```
   ```ps1
   Import-Module LyncOnlineConnector
   $cssess=New-CsOnlineSession -Credential $cred
   Import-PSSession $cssess -AllowClobber
   ```
   ```

2. To enable your Surface Hub account for Skype for Business Server, run this cmdlet:

   ```
   ```ps1
   Enable-CsMeetingRoom -Identity 'HUB01@contoso.com' -RegistrarPool 'sippoolbl20a04.infra.lync.com' -
   SipAddressType UserPrincipalName
   ```
   ```

   If you aren't sure what value to use for the `RegistrarPool` parameter in your environment, you can get the value from an existing Skype for Business user using this cmdlet:

   ```
   ```ps1
   Get-CsOnlineUser -Identity ‘HUB01@contoso.com’| fl *registrarpool*
   ```
   ```

3. Assign Skype for Business license to your Surface Hub account.

   Once you've completed the preceding steps to enable your Surface Hub account in Skype for Business Online, you need to assign a license to the Surface Hub. Using the O365 administrative portal, assign either a Skype for Business Online (Plan 2) or a Skype for Business Online (Plan 3) to the device.

   - Login as a tenant administrator, open the O365 Administrative Portal, and click on the Admin app.
   - Click on **Users and Groups** and then **Add users, reset passwords, and more**.
   - Click the Surface Hub account, and then click the pen icon to edit the account information.

- Click **Licenses**.
- In **Assign licenses**, select Skype for Business (Plan 2) or Skype for Business (Plan 3), depending on your licensing and Enterprise Voice requirements. You'll have to use a Plan 3 license if you want to use Enterprise Voice on your Surface Hub.
- Click **Save**.

> **NOTE**
> You can also use the Windows Azure Active Directory Module for Windows Powershell to run the cmdlets needed to assign one of these licenses, but that's not covered here.

For validation, you should be able to use any Skype for Business client (PC, Android, etc.) to sign in to this account.

Skype for Business on-prem

To run this cmdlet, you will need to connect to one of the Skype front-ends. Open the Skype PowerShell and run:

```
Enable-CsMeetingRoom -Identity 'HUB01@contoso.com' -RegistrarPool registrarpoolfqdn -SipAddressType
UserPrincipalName
```

Skype for Business hybrid

If your organization has set up hybrid connectivity between Skype for Business Server and Skype for Business Online, the guidance for creating accounts differs from a standard Surface Hub deployment.

The Surface Hub requires a Skype account of the type `meetingroom`, while a normal user would use a user type account in Skype. If your Skype server is set up for hybrid where you might have users on the local Skype server as well as users hosted in Office 365, you might run into a few issues when trying to create a Surface Hub account.

In a hybrid Skype environment, you have to create the user on-prem first, then move the user to the cloud. This means that your user is present in both environments (which makes SIP routing possible). The move from on-prem to online is done via the Move-CsUser cmdlet which can only be used against user type accounts, not meetingroom type accounts. Because of this, you will not be able to move a Surface Hub account that has a meetingroom type of account. You might think of using the Move-CsMeetingRoom cmdlet, unfortunately this will not work between the on-prem Skype server and Office 365 - it only works across on-prem Skype pools.

To have a functional Surface Hub account in a Skype hybrid configuration, create the Skype account as a normal user type account, instead of creating the account as a meetingroom. Enable the account on the on-prem Skype server first:

```
Enable-CsUser -Identity 'HUB01@contoso.com' -RegistrarPool "registrarpoolfqdn" -SipAddressType
UserPrincipalName
```

After the Surface Hub account is enabled for Skype for Business on-premises, you can keep the account on-premises or you can move the Surface Hub account to Office 365, using the Move-CsUser cmdlet. Learn more about moving a Skype user to Office 365.

Exchange online

Use this procedure if you use Exchange online.

1. Create an email account in Office 365.

   Start a remote PowerShell session on a PC and connect to Exchange. Be sure you have the right permissions set to run the associated cmdlets.

```
Set-ExecutionPolicy Unrestricted
$cred=Get-Credential -Message "Please use your Office 365 admin credentials"
$sess= New-PSSession -ConfigurationName Microsoft.Exchange -ConnectionUri
https://outlook.office365.com/ps1-liveid/ -Credential $cred -Authentication Basic -AllowRedirection
Import-PSSession $sess
```

2. Set up mailbox.

   After establishing a session, you'll either create a new mailbox and enable it as a RoomMailboxAccount, or change the settings for an existing room mailbox. This will allow the account to authenticate into the Surface Hub.

   If you're changing an existing resource mailbox:

   ```
   Set-Mailbox -Identity 'HUB01' -EnableRoomMailboxAccount $true -RoomMailboxPassword (ConvertTo-
   SecureString -String <password> -AsPlainText -Force)
   ```

   If you're creating a new resource mailbox:

   ```
   New-Mailbox -MicrosoftOnlineServicesID 'HUB01@contoso.com' -Alias HUB01 -Name "Hub-01" -Room -
   EnableRoomMailboxAccount $true -RoomMailboxPassword (ConvertTo-SecureString -String <password> -
   AsPlainText -Force)
   ```

3. Create Exchange ActiveSync policy.

   After setting up the mailbox, you will need to either create a new Exchange ActiveSync policy, or use a compatible existing policy.

   Surface Hubs are only compatible with device accounts that have an ActiveSync policy where the **PasswordEnabled** property is set to False. If this isn't set properly, then Exchange services on the Surface Hub (mail, calendar, and joining meetings), will not be enabled.

   If you haven't created a compatible policy yet, use the following cmdlet—this one creates a policy called "Surface Hubs". Once it's created, you can apply the same policy to other device accounts.

   ```
   $easPolicy = New-MobileDeviceMailboxPolicy -Name "SurfaceHubs" -PasswordEnabled $false
   ```

   Once you have a compatible policy, then you will need to apply the policy to the device account. However, policies can only be applied to user accounts and not resource mailboxes. You need to convert the mailbox into a user type, apply the policy, and then convert it back into a mailbox—you may need to re-enable it and set the password again too.

   ```
   Set-Mailbox 'HUB01@contoso.com' -Type Regular
   Set-CASMailbox 'HUB01@contoso.com' -ActiveSyncMailboxPolicy $easPolicy.id
   Set-Mailbox 'HUB01@contoso.com' -Type Room
   $credNewAccount = Get-Credential -Message "Please provide the Surface Hub username and password"
   Set-Mailbox 'HUB01@contoso.com' -RoomMailboxPassword $credNewAccount.Password -EnableRoomMailboxAccount
   $true
   ```

4. Set Exchange properties.

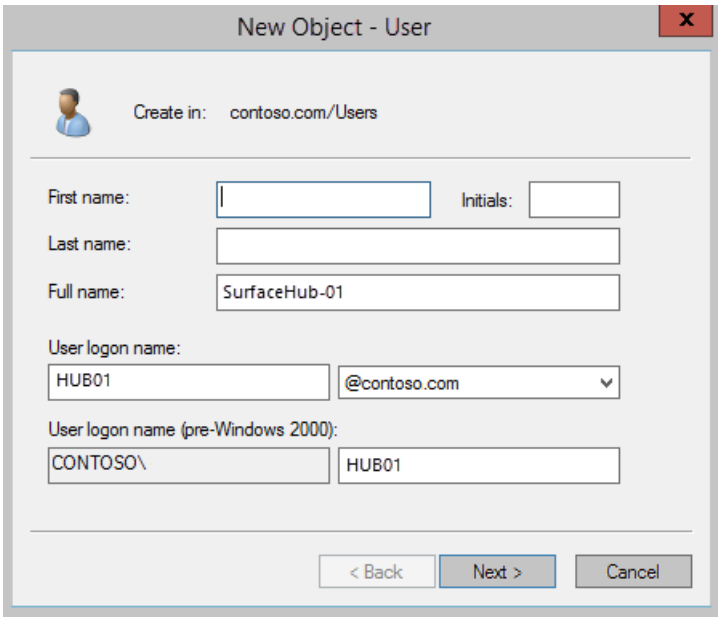
   Various Exchange properties must be set on the device account to improve the meeting experience. You can see which properties need to be set in the Exchange properties section.

```
Set-CalendarProcessing -Identity 'HUB01@contoso.com' -AutomateProcessing AutoAccept -
AddOrganizerToSubject $false –AllowConflicts $false –DeleteComments $false -DeleteSubject $false -
RemovePrivateProperty $false
Set-CalendarProcessing -Identity 'HUB01@contoso.com' -AddAdditionalResponse $true -AdditionalResponse
"This is a Surface Hub room!"
```

5. Add email address for your on-prem domain account.

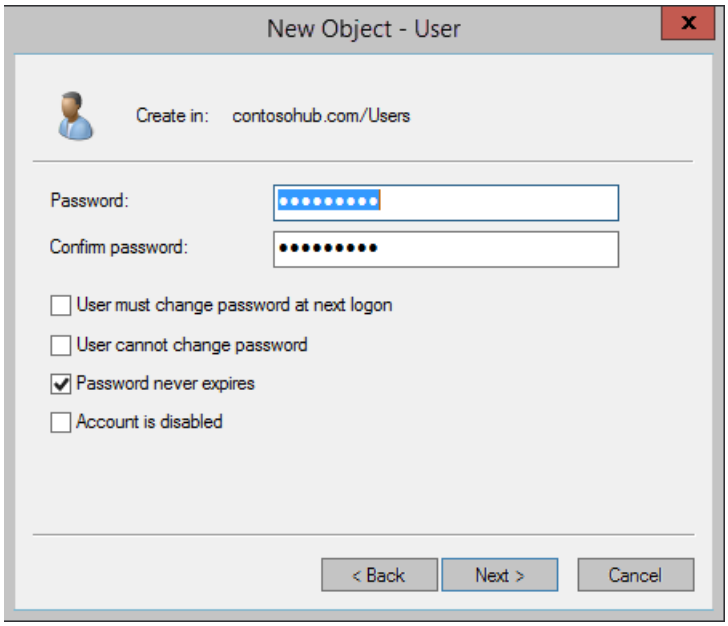
   For this procedure, you'll be using AD admin tools to add an email address for your on-prem domain account.

   - In **Active Directory Users and Computers** AD tool, right-click on the folder or Organizational Unit that your Surface Hub accounts will be created in, click **New**, and **User**.
   - Type the display name from the previous cmdlet into the **Full name** box, and the alias into the **User logon name** box. Click **Next**.

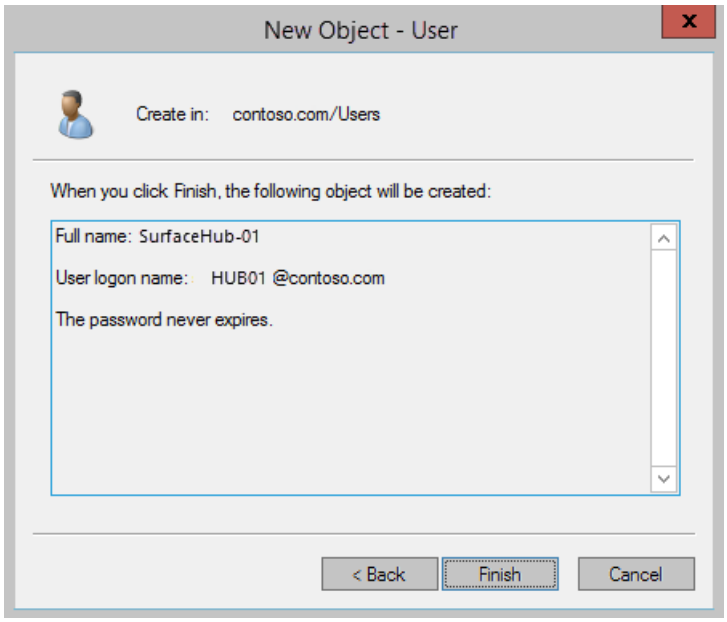


   - Type the password for this account. You'll need to retype it for verification. Make sure the **Password never expires** checkbox is the only option selected.

     Important Selecting **Password never expires** is a requirement for Skype for Business on the Surface Hub. Your domain rules may prohibit passwords that don't expire. If so, you'll need to create an exception for each Surface Hub device account.



- Click **Finish** to create the account.



6. Run directory synchronization.

   After you've created the account, run a directory synchronization. When it's complete, go to the users page and verify that the two accounts created in the previous steps have merged.

7. Connect to Azure AD.

   You need to connect to Azure AD to apply some account settings. You can run this cmdlet to connect.

   ```
   Connect-MsolService -Credential $cred
   ```

8. Assign an Office 365 license.

   The device account needs to have a valid Office 365 (O365) license, or Exchange and Skype for Business will not work. If you have the license, you need to assign a usage location to your device account—this determines what license SKUs are available for your account.

   Next, you can use `Get-MsolAccountSku` to retrieve a list of available SKUs for your O365 tenant.

   Once you list out the SKUs, you can add a license using the `Set-MsolUserLicense` cmdlet. In this case,

`$strLicense` is the SKU code that you see (for example, *contoso:STANDARDPACK*).

```
Set-MsolUser -UserPrincipalName 'HUB01@contoso.com' -UsageLocation 'US'
Get-MsolAccountSku
Set-MsolUserLicense -UserPrincipalName 'HUB01@contoso.com' -AddLicenses $strLicense
```

Next, you enable the device account with Skype for Business Online, Skype for Business on-prem, or Skype for Business hybrid.

Skype for Business Online

In order to enable Skype for Business, your environment will need to meet the following prerequisites:

- You'll need to have Lync Online (Plan 2) or higher in your O365 plan. The plan needs to support conferencing capability.
- If you need Enterprise Voice (PSTN telephony) using telephony service providers for the Surface Hub, you need Lync Online (Plan 3).
- Your tenant users must have Exchange mailboxes (at least one Exchange mailbox in the tenant is required).
- Your Surface Hub account does require a Lync Online (Plan 2) or Lync Online (Plan 3) license, but it does not require an Exchange Online license.

1. Start by creating a remote PowerShell session to the Skype for Business online environment from a PC.

   ```
   Import-Module LyncOnlineConnector
   $cssess=New-CsOnlineSession -Credential $cred
   Import-PSSession $cssess -AllowClobber
   ```

2. To enable your Surface Hub account for Skype for Business Server, run this cmdlet:

   ```
   Enable-CsMeetingRoom -Identity 'HUB01@contoso.com' -RegistrarPool
   'sippoolbl20a04.infra.lync.com' -SipAddressType UserPrincipalName
   ```

   If you aren't sure what value to use for the `RegistrarPool` parameter in your environment, you can get the value from an existing Skype for Business user using this cmdlet:

   ```
   Get-CsOnlineUser -Identity 'HUB01@contoso.com'| fl *registrarpool*
   ```

3. Assign Skype for Business license to your Surface Hub account

   Once you've completed the preceding steps to enable your Surface Hub account in Skype for Business Online, you need to assign a license to the Surface Hub. Using the O365 administrative portal, assign either a Skype for Business Online (Plan 2) or a Skype for Business Online (Plan 3) to the device.

   - Sign in as a tenant administrator, open the O365 Administrative Portal, and click on the Admin app.
   - Click on **Users and Groups** and then **Add users, reset passwords, and more**.
   - Click the Surface Hub account, and then click the pen icon to edit the account information.
   - Click **Licenses**.
   - In **Assign licenses**, select Skype for Business (Plan 2) or Skype for Business (Plan 3), depending on your licensing and Enterprise Voice requirements. You'll have to use a Plan 3 license if you want to use Enterprise Voice on your Surface Hub.

- Click **Save**.

> **NOTE**
> You can also use the Windows Azure Active Directory Module for Windows PowerShell to run the cmdlets needed to assign one of these licenses, but that's not covered here.

For validation, you should be able to use any Skype for Business client (PC, Android, etc) to sign in to this account.

Skype for Business on-prem

To run this cmdlet, you will need to connect to one of the Skype front-ends. Open the Skype PowerShell and run:

```
Enable-CsMeetingRoom -Identity 'HUB01@contoso.com' -RegistrarPool registrarpoolfqdn -SipAddressType
UserPrincipalName
```

Skype for Business hybrid

If your organization has set up hybrid connectivity between Skype for Business Server and Skype for Business Online, the guidance for creating accounts differs from a standard Surface Hub deployment.

The Surface Hub requires a Skype account of the type *meetingroom*, while a normal user would use a *user* type account in Skype. If your Skype server is set up for hybrid where you might have users on the local Skype server as well as users hosted in Office 365, you might run into a few issues when trying to create a Surface Hub account.

In a hybrid Skype environment, you have to create the user on-prem first, then move the user to the cloud. This means that your user is present in both environments (which makes SIP routing possible). The move from on-prem to online is done via the Move-CsUser cmdlet which can only be used against user type accounts, not meetingroom type accounts. Because of this, you will not be able to move a Surface Hub account that has a meetingroom type of account. You might think of using the Move-CsMeetingRoom cmdlet, unfortunately this will not work between the on-prem Skype server and Office 365 - it only works across on-prem Skype pools.

In order to have a functional Surface Hub account in a Skype hybrid configuration, create the Skype account as a normal user type account, instead of creating the account as a meetingroom. First follow the Exchange steps - either online or on-prem - and, instead of enabling the user for Skype for Business Online as described, enable the account on the on-prem Skype server:

```
Enable-CsUser -Identity 'HUB01@contoso.com' -RegistrarPool "registrarpoolfqdn" -SipAddressType
UserPrincipalName
```

After the Surface Hub account is enabled for Skype for Business on-premises, you can keep the account on-premises or you can move the Surface Hub account to Office 365, using the Move-CsUser cmdlet. Learn more about moving a Skype user to Office 365.

Online or hybrid deployment using Skype Hybrid Voice environment (Surface Hub)

5/4/2017 • 4 min to read • Edit Online

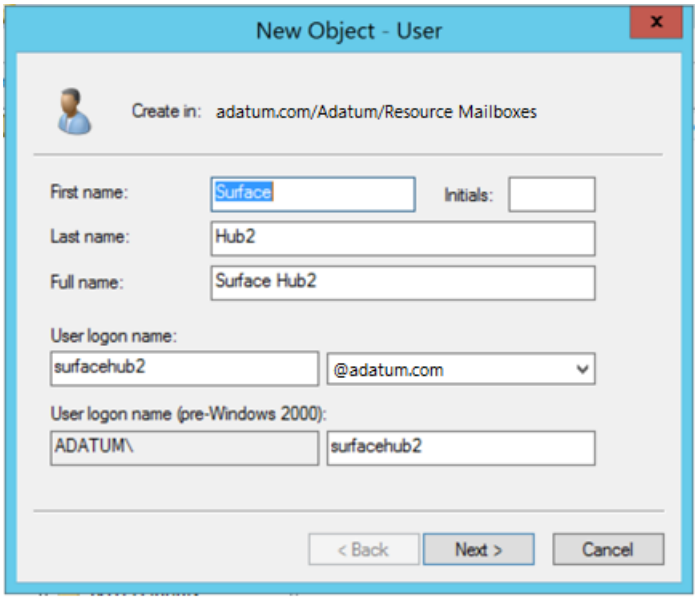
This topic explains how to enable Skype for Business Cloud PBX with on-premises Public Switched Telephone Network (PSTN) connectivity via Cloud Connector Edition or Skype for Business 2015 pool. In this option. your Skype for Business home pools and Exchange servers are in the cloud, and are connected by PSTN via an on-premises pool running Skype for Business 2015 or Cloud Connector edition. Learn more about different Cloud PBX options.

If you deployed Skype for Business Cloud PBX with one of the hybrid voice options, follow the steps below to enable the room account for Surface Hub. It is important to create a regular user account first, assign all hybrid voice options and phone numbers, and then convert the account to a room account. If you do not follow this order, you will not be able to assign a hybrid phone number.

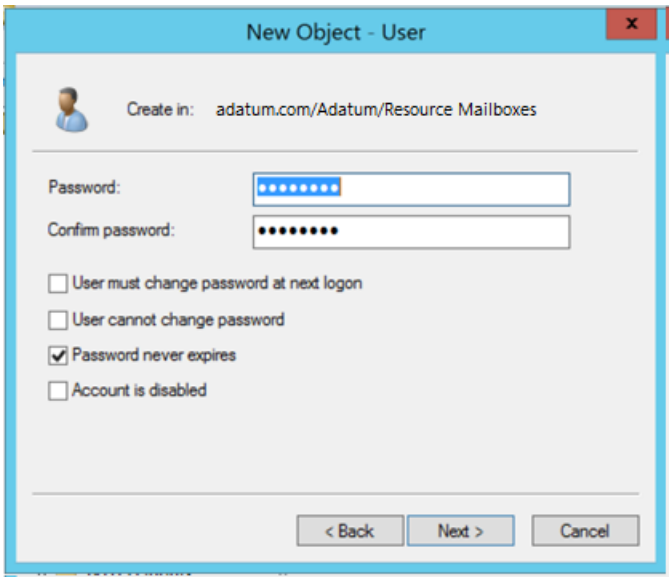
WARNING

If you create an account before configuration of Hybrid voice (you run Enable-CSMeetingRoom command), you will not be able to configure required hybrid voice parameters. In order to configure hybrid voice parameters for a previously configured account or to reconfigure a phone number, delete the E5 or E3 + Cloud PBX add-on license, and then follow the steps below, starting at step 3.

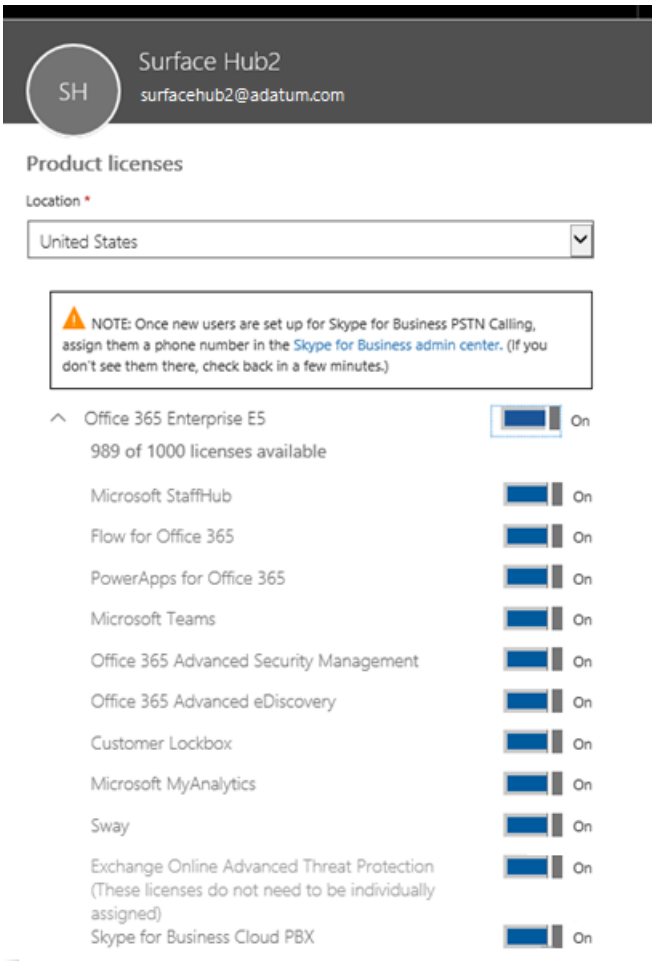
1. Create a new user account for Surface Hub. This example uses **surfacehub2@adatum.com**. The account can be created in local Active Directory and synchronized to the cloud, or created directly in the cloud.



2. Select **Password Never Expires**. This is important for a Surface Hub device.



3. In Office 365, add **E5** license or **E3 and Cloud PBX** add-on to the user account created for the room. This is required for Hybrid Voice to work.



4. Wait approximately 15 minutes until the user account for the room appears in Skype for Business Online.

5. After the user account for room is created in Skype for Business Online, enable it for Hybrid Voice in Skype for Business Remote PowerShell by running the following cmdlet:

```
Set-csuser surfacehub2@adatum.com EnterpriseVoiceEnabled $true -HostedVoiceMail $true -onpremlineuri
tel:+15005000102
```

6. Validate Hybrid Voice call flow by placing test calls from the Surface Hub.

7. Start a remote PowerShell session on a PC and connect to Exchange by running the following cmdlets.

```
Set-ExecutionPolicy Unrestricted
$cred=Get-Credential -Message "Please use your Office 365 admin credentials"
$sess= New-PSSession -ConfigurationName Microsoft.Exchange -ConnectionUri
https://outlook.office365.com/ps1-liveid/ -Credential $cred -Authentication Basic -AllowRedirection
Import-PSSession $sess
```

8. After establishing a session, modify the user account for the room to enable it as a **RoomMailboxAccount** by running the following cmdlets. This allows the account to authenticate with Surface Hub.

```
Set-Mailbox surfacehub2@adatum.com -Type Room
Set-Mailbox surfacehub2@adatum.com -EnableRoomMailboxAccount $true -RoomMailboxPassword (ConvertTo-
SecureString -String <password> -AsPlainText -Force)
```

9. After setting up the mailbox, you will need to either create a new Exchange ActiveSync policy, or use a compatible existing policy.

   Surface Hubs are only compatible with device accounts that have an ActiveSync policy where the **PasswordEnabled** property is set to **False**. If this isn't set properly, then Exchange services on the Surface Hub (mail, calendar, and joining meetings), will not be enabled.

   If you haven't created a compatible policy yet, use the following cmdlet (this one creates a policy called "Surface Hubs"). After it's created, you can apply the same policy to other device accounts.

```
$easPolicy = New-MobileDeviceMailboxPolicy -Name “SurfaceHubs” -PasswordEnabled $false
```

   After you have a compatible policy, then you will need to apply the policy to the device account. However, policies can only be applied to user accounts and not resource mailboxes. Run the following cmdlets to convert the mailbox into a user type, apply the policy, and then convert it back into a mailbox (you may need to re-enable the account and set the password again).

```
Set-Mailbox surfacehub2@adatum.com -Type Regular
Set-CASMailbox surfacehub2@adatum.com -ActiveSyncMailboxPolicy $easPolicy.id
Set-Mailbox surfacehub2@adatum.com -Type Room
$credNewAccount = Get-Credential -Message "Please provide the Surface Hub username and password"
Set-Mailbox surfacehub2@adatum.com -RoomMailboxPassword $credNewAccount.Password -
EnableRoomMailboxAccount $true
```

10. Various Exchange properties must be set on the device account to improve the meeting experience. You can see which properties can be set in Exchange properties. The following cmdlets provide an example of setting Exchange properties.

```
Set-CalendarProcessing surfacehub2@adatum.com -AutomateProcessing AutoAccept -AddOrganizerToSubject
$false –AllowConflicts $false –DeleteComments $false -DeleteSubject $false -RemovePrivateProperty $false
Set-CalendarProcessing surfacehub2@adatum.com -AddAdditionalResponse $true -AdditionalResponse "This is
a Surface Hub room!"
```

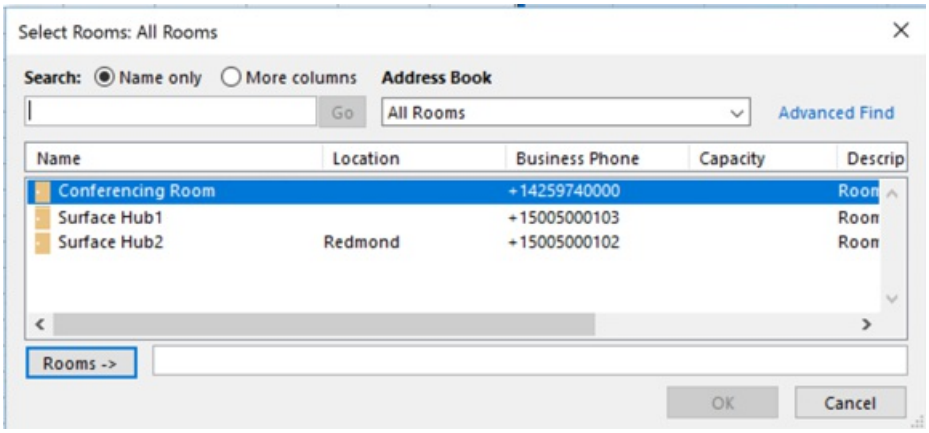
11. Enable the mailbox as a meeting device in Skype for Business Online. Run the following cmdlet which enables the acount as a meeting device.

```
Get-CsTenant | select registrarpool
Enable-CsMeetingRoom surfacehub2@adatum.com -RegistrarPool  'sippoolbl20a04.infra.lync.com' -
SipAddressType UserPrincipalName
```

As a result of running this cmdlet, users will be asked if they are in a meeting room, as shown in the following image. **Yes** will mute the microphone and speaker.

At this moment the room account is fully configured, including Hybrid Voice. If you use Skype on-premises, you can configure additional attributes, like description, location, etc., on-premises. If you create a room in Skype Online, these parameters can be set online.

In the following image, you can see how the device appears to users.



Create a device account using UI (Surface Hub)

5/4/2017 • 11 min to read • Edit Online

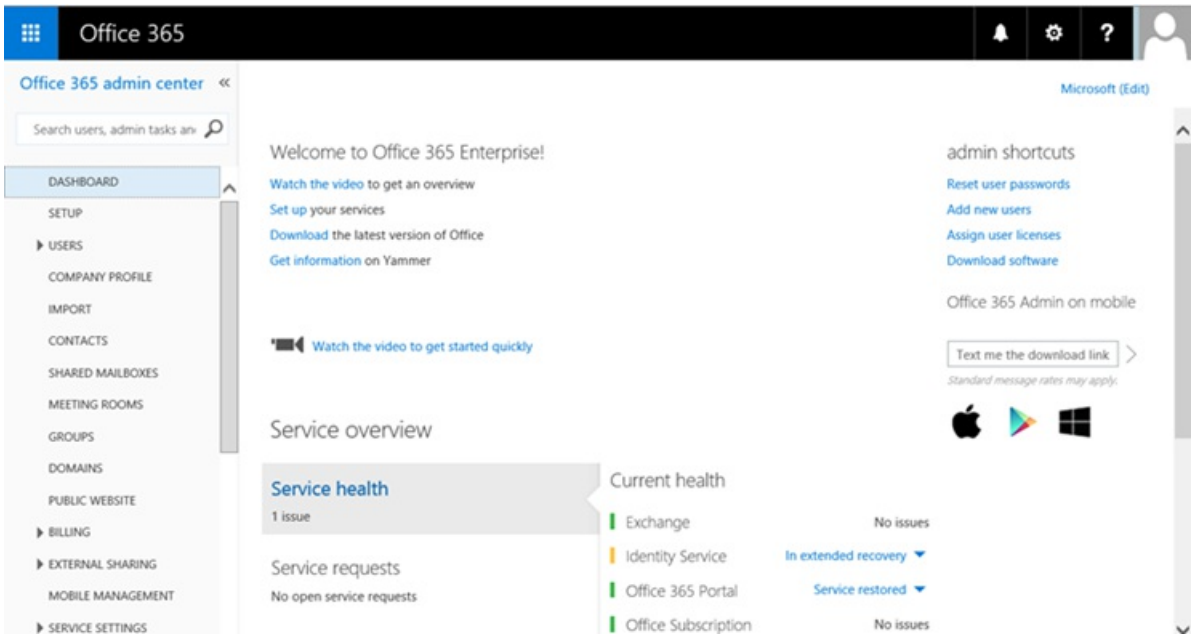
If you prefer to use a graphical user interface, you can create a device account for your Microsoft Surface Hub with either the Office 365 UI or the Exchange Admin Center.

Create a device account using Office 365

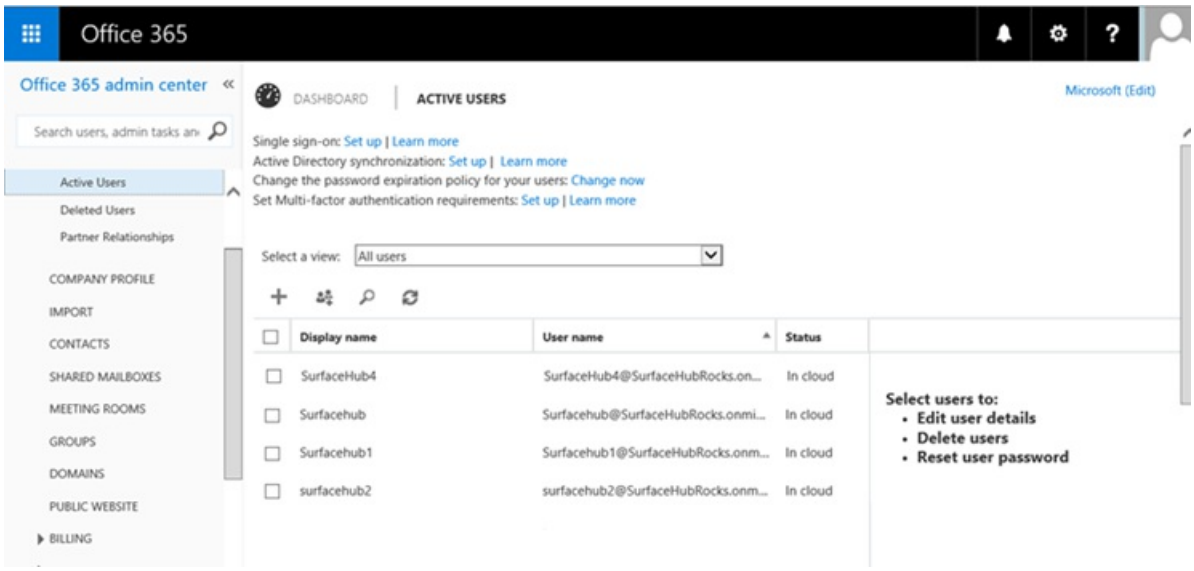
1. Create the account in the Office 365 Admin Center.
2. Create a mobile device mailbox (ActiveSync) policy from the Microsoft Exchange Admin Center.
3. Use PowerShell to complete device account creation.
4. Use PowerShell to configure Exchange properties of the account.
5. Enable the account with Skype for Business.

Create the account in the Office 365 Admin Center

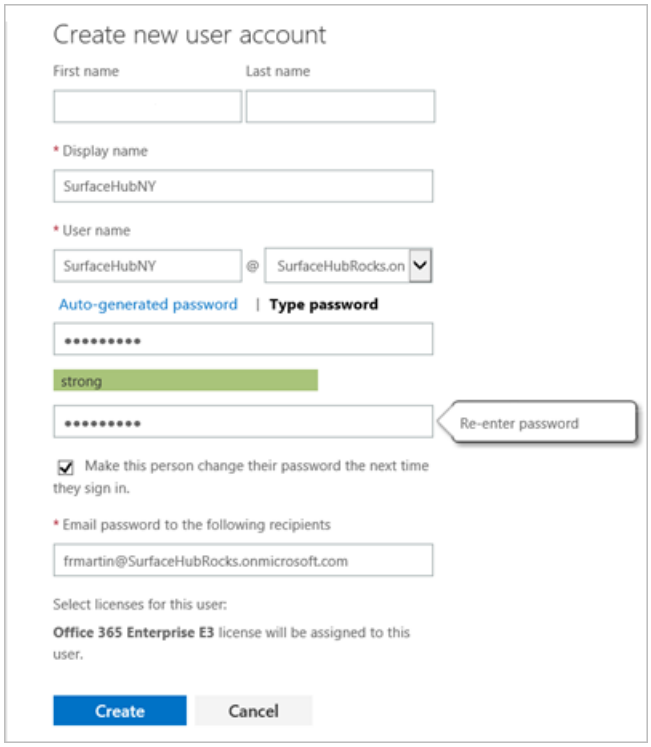
1. Sign in to Office 365 by visiting http://portal.office.com
2. Provide the admin credentials for your Office 365 tenant. This will take you to your Office 365 Admin Center.



3. Once you are at the Office 365 Admin Center, navigate to **Users** in the left panel, and then click **Active Users**.

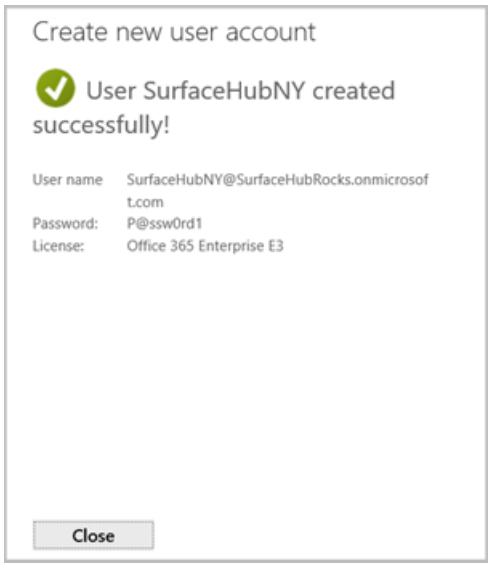


4. On the controls above the list of users, click **+** to create a new user. You'll need to enter a **Display name**, **User name**, **Password** and an email address for the recipient of the password. Optionally you can change the password manually, but we recommend that you use the auto-generated option. You also need to assign this account a license that gives the account access to Exchange and Skype for Business services.

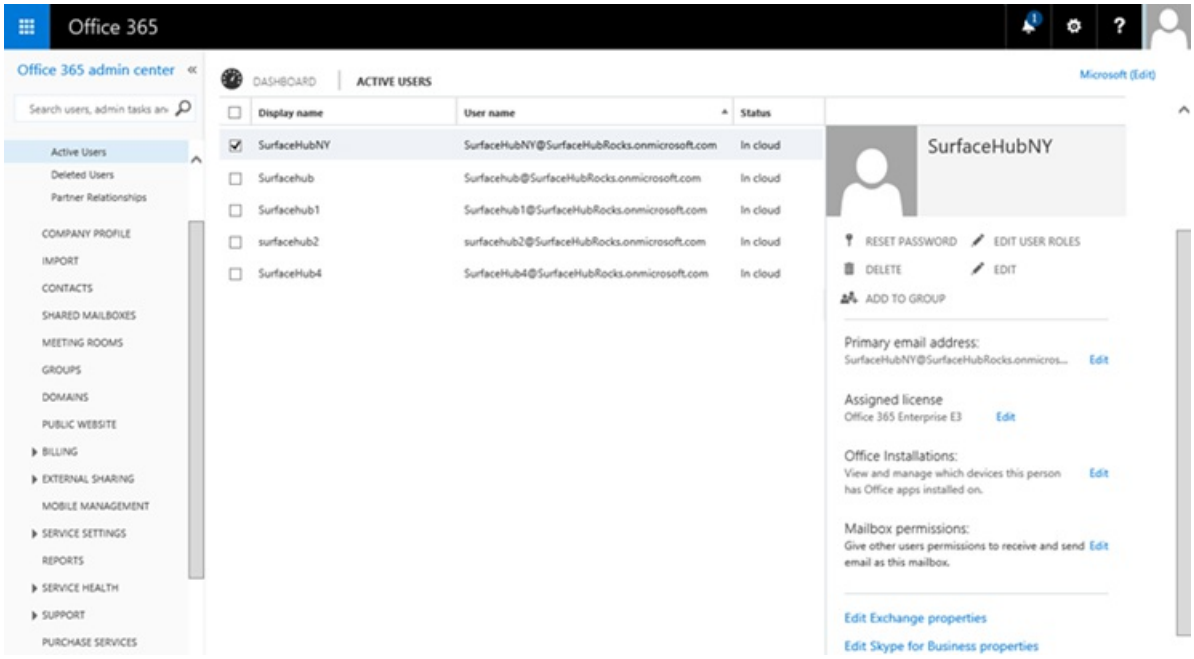


   Click **Create**.

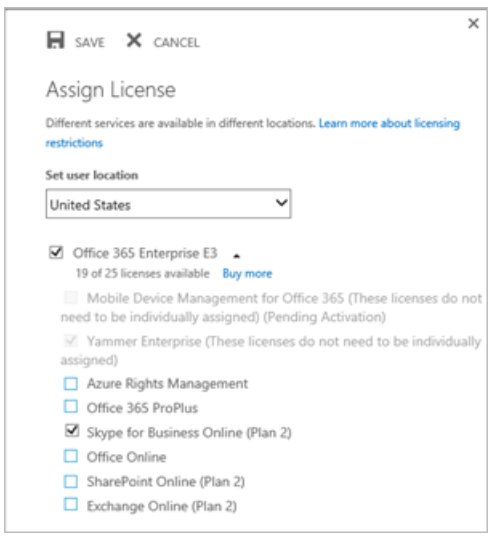
5. Once the account has been successfully created, click **Close** on the resulting dialog box, and you will see the admin center Active Users list again.



6. Select the user you just created from the **Active Users** list. You need to disable the Skype for Business license, because you can't create a Skype Meeting Room with this option.



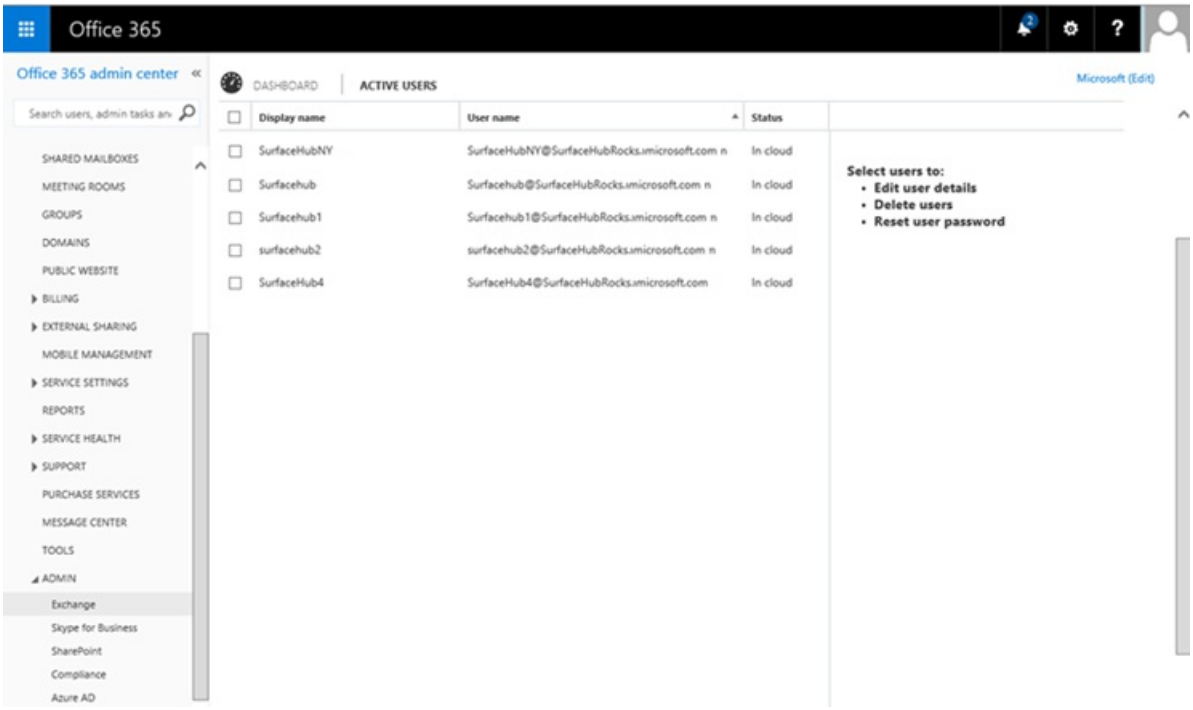
In the right panel you can see the account properties and several optional actions. The process so far has created a regular Skype account for this user, which you need to disable. Click **Edit** for the **Assigned license** section, then click the dropdown arrow next to the license to expand the details.



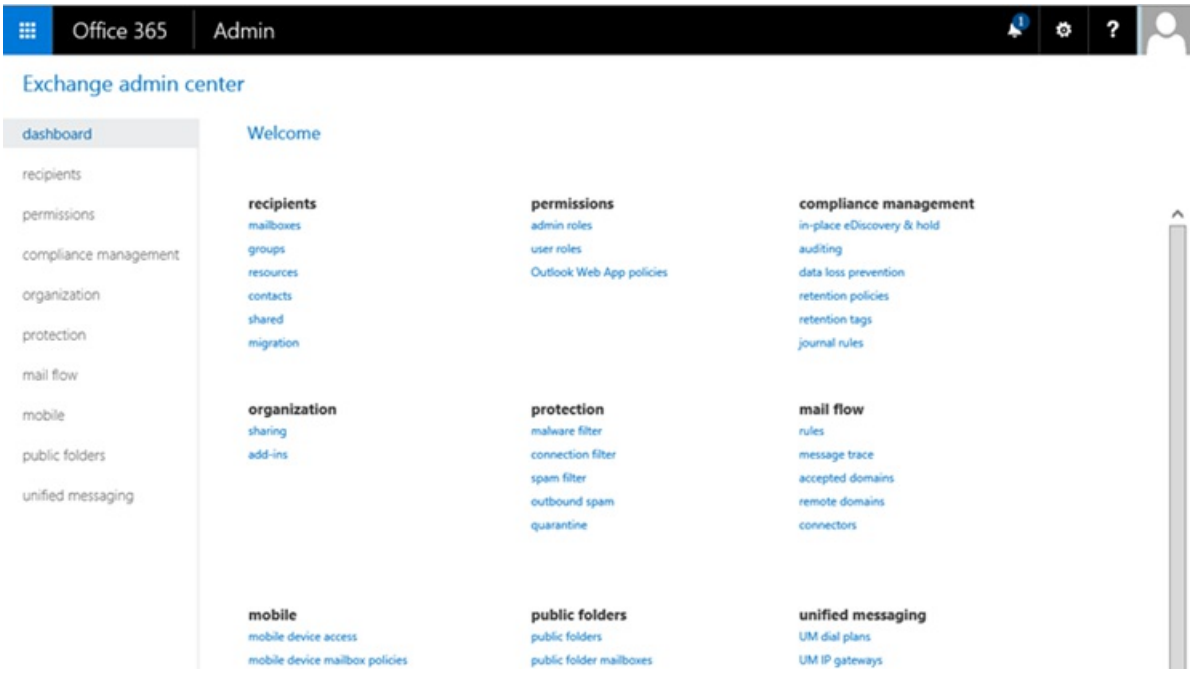
From the list, select **Skype for Business Online (Plan 2)**, and then click **SAVE**. The license may vary depending on your organization (for example, you might have Plan 2, or Plan 3).

Create a mobile device mailbox (ActiveSync) policy from the Exchange Admin Center

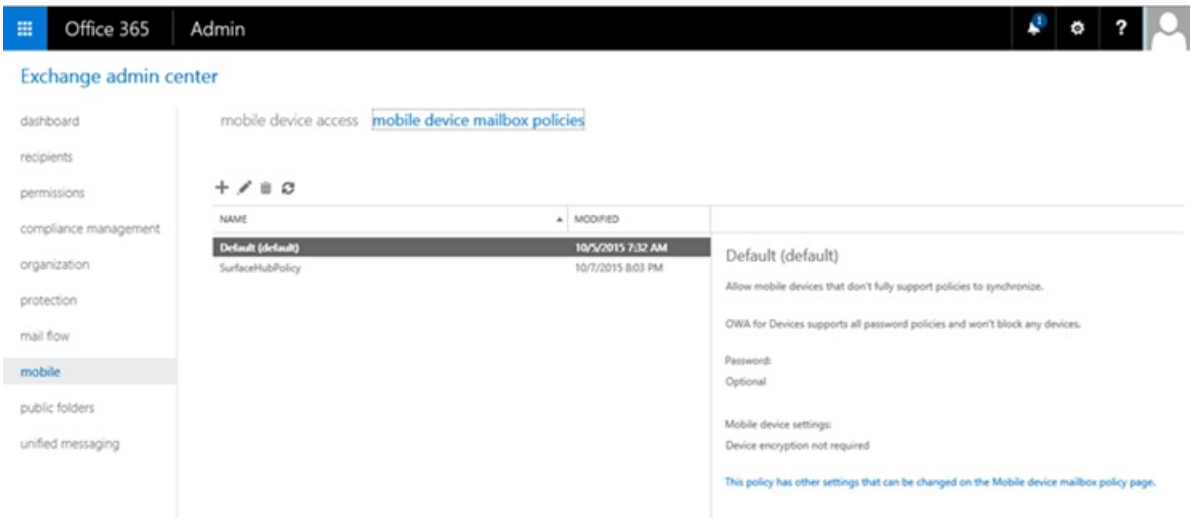
1. In the Office 365 Admin Center's left panel, click **ADMIN**, and then click **Exchange**.



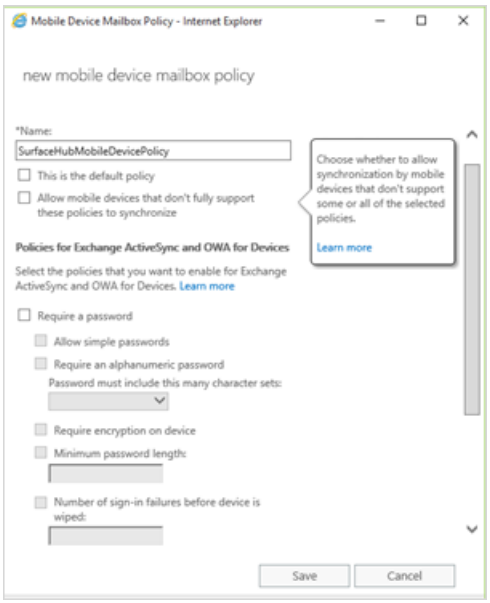
2. This will open another tab on your browser to take you to the Exchange Admin Center, where you can create and set the Mailbox Setting for Surface Hub.



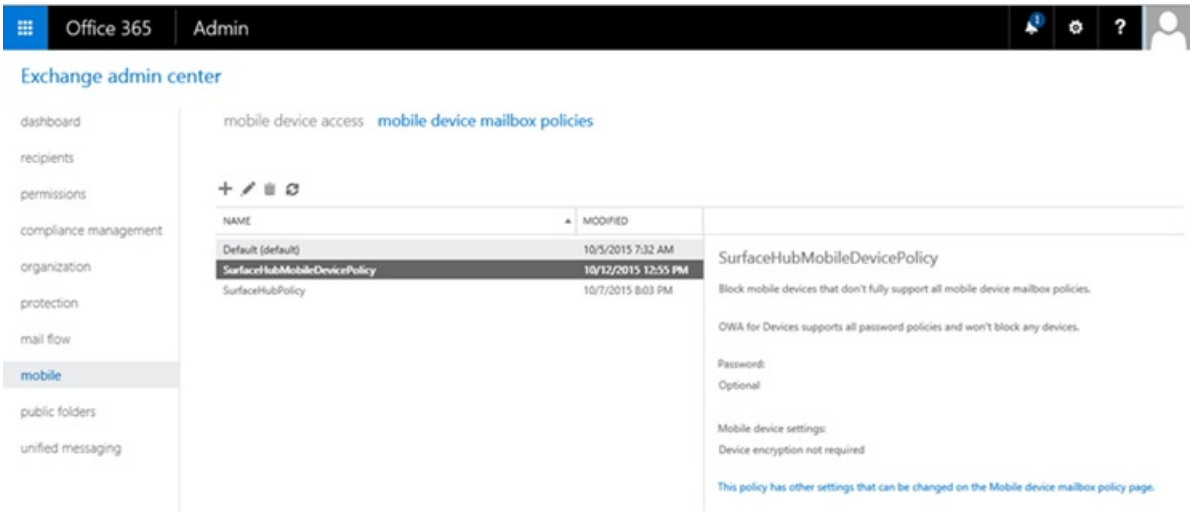
3. To create a Mobile Device Mailbox Policy, click **Mobile** from the left panel and then click **Mobile device mailbox policies**. Surface Hubs require an account with a mobile device mailbox policy that does not require a password, so if you already have an existing policy that matches this requirement, you can apply that policy to the account. Otherwise use the following steps to create a new one to be used only for Surface Hub device accounts.



4. To create a New Surface Hub mobile device mailbox policy, click the **+** button from the controls above the list of policies to add a new policy. For the name, provide a name that will help you distinguish this policy from other device accounts (for example, *SurfaceHubDeviceMobilePolicy*). Make sure the policy does not require a password for the devices assigned to, so make sure **Require a Password** remains unchecked, then click **Save**.



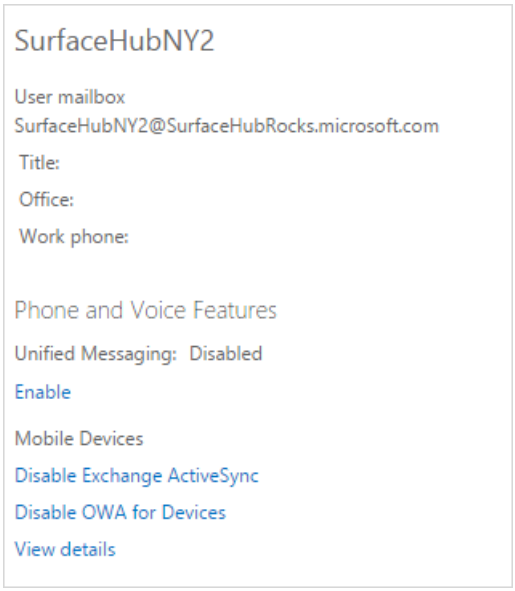
5. After you have created the new mobile device mailbox policy, go back to the **Exchange Admin Center** and you will see the new policy listed.



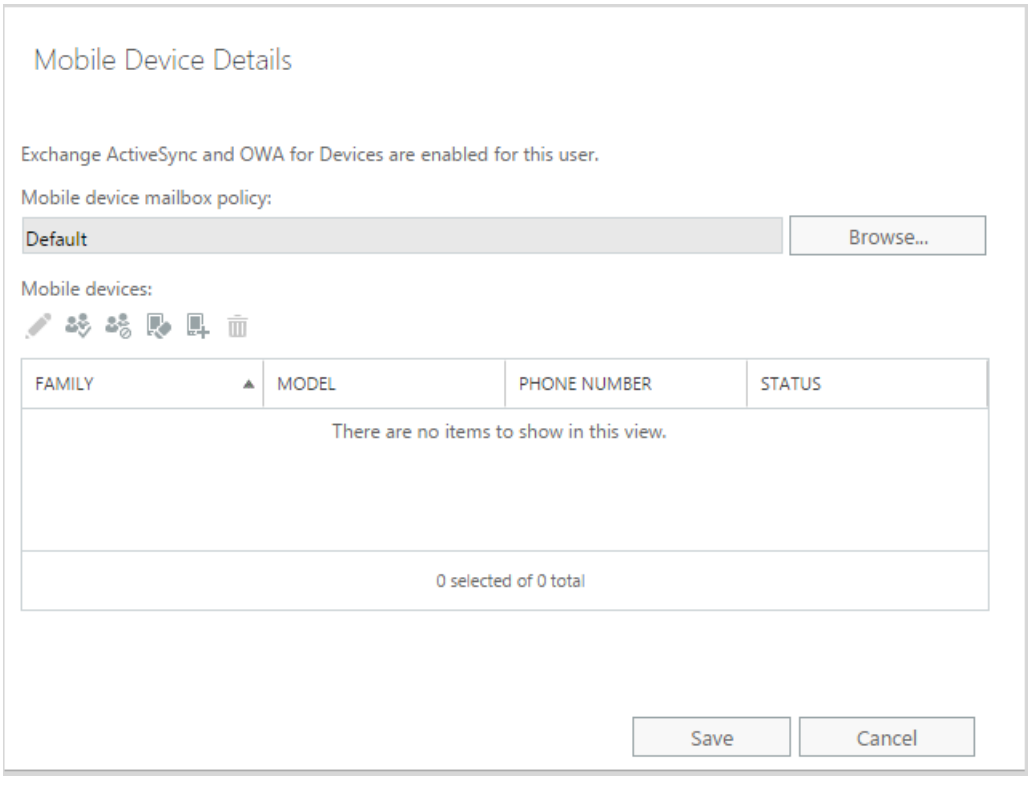
6. Now, to apply the ActiveSync policy without using PowerShell, you can do the following: In the EAC, click

Recipients > **Mailboxes** and then select a mailbox.

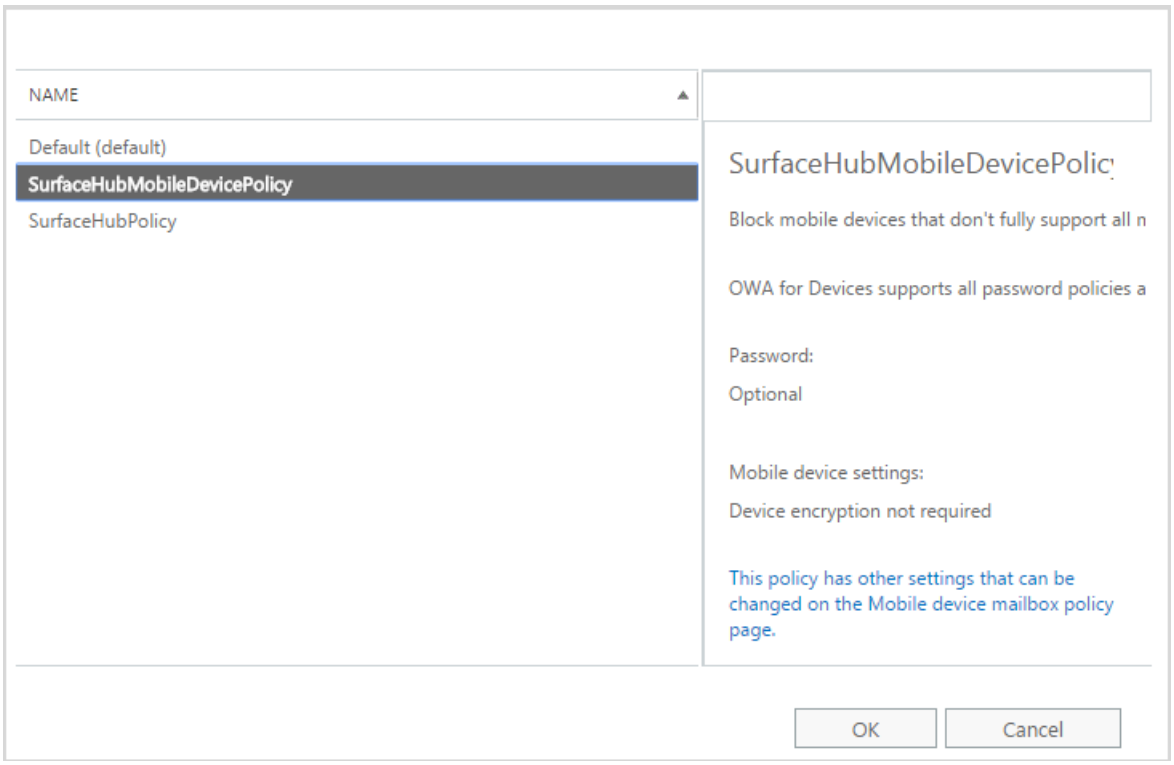
7. In the Details pane, scroll to **Phone and Voice Features** and click **View details** to display the **Mobile Device Details** screen.



8. The mobile device mailbox policy that's currently assigned is displayed. To change the mobile device mailbox policy, click **Browse**.



9. Choose the appropriate mobile device mailbox policy from the list, click **OK** and then click **Save**.



Use PowerShell to complete device account creation

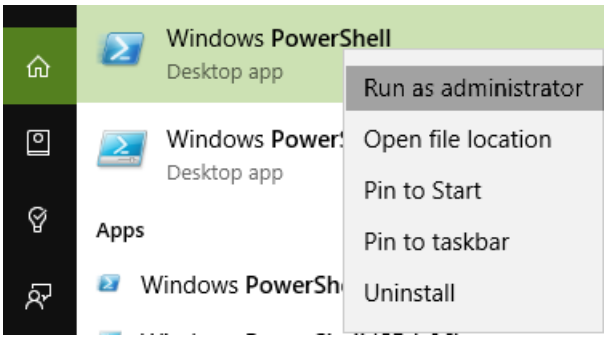
From here on, you'll need to finish the account creation process using PowerShell to set up some configuration.

In order to run cmdlets used by these PowerShell scripts, the following must be installed for the admin PowerShell console:

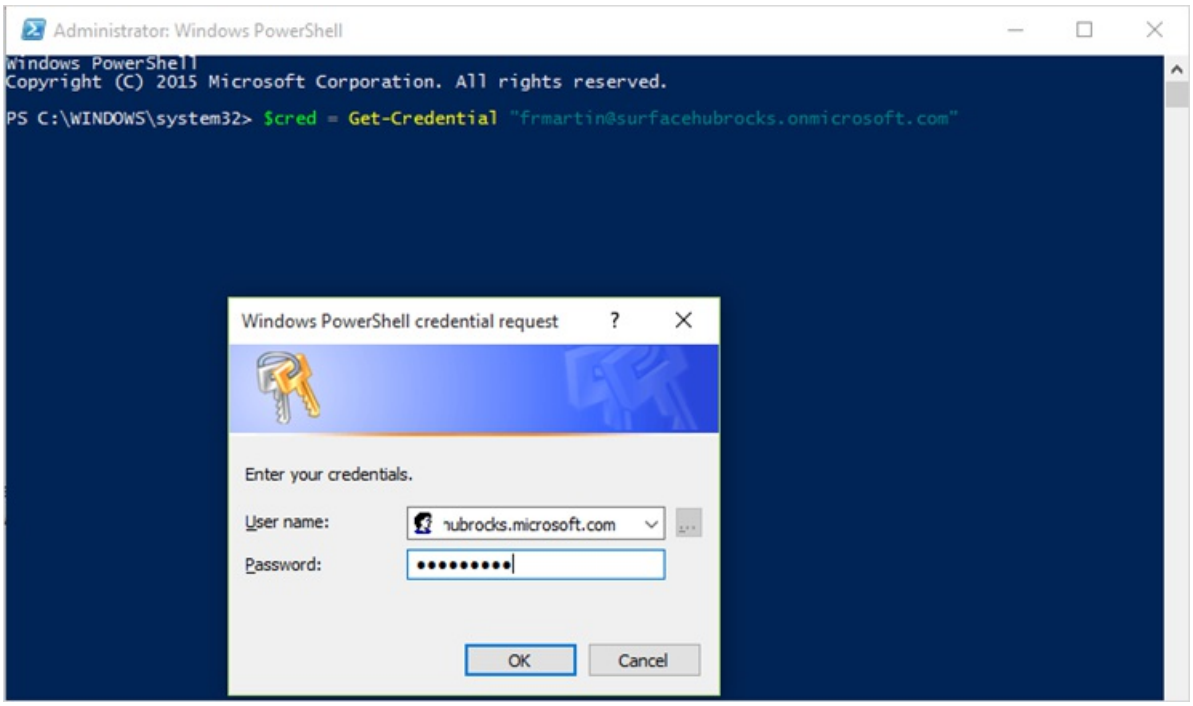
- Microsoft Online Services Sign-In Assistant for IT Professionals BETA
- Windows Azure Active Directory Module for Windows PowerShell
- Skype for Business Online, Windows PowerShell Module

Connecting to online services

1. Run Windows PowerShell as Administrator.

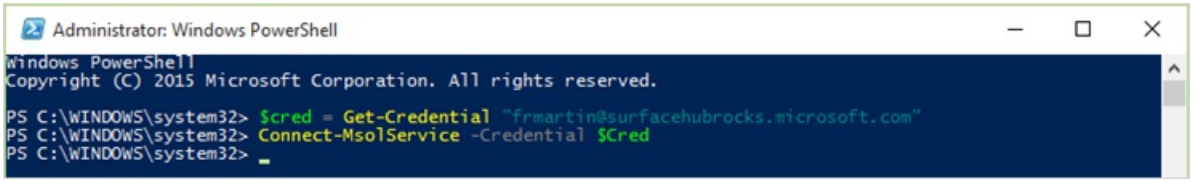


2. Create a Credentials object, then create a new session that connects to Skype for Business Online, and provide the global tenant administrator account, then click **OK**.



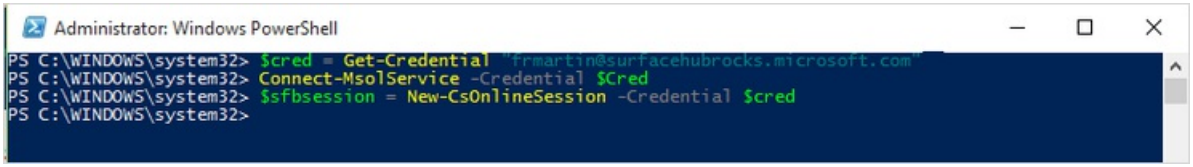
3. To connect to Microsoft Online Services, run:

```
Connect-MsolService -Credential $Cred
```



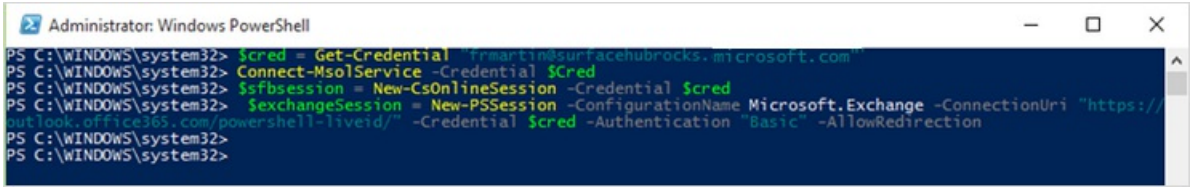
4. Now to connect to Skype for Business Online Services, run:

```
$sfbsession = New-CsOnlineSession -Credential $cred
```



5. Finally, to connect to Exchange Online Services, run:

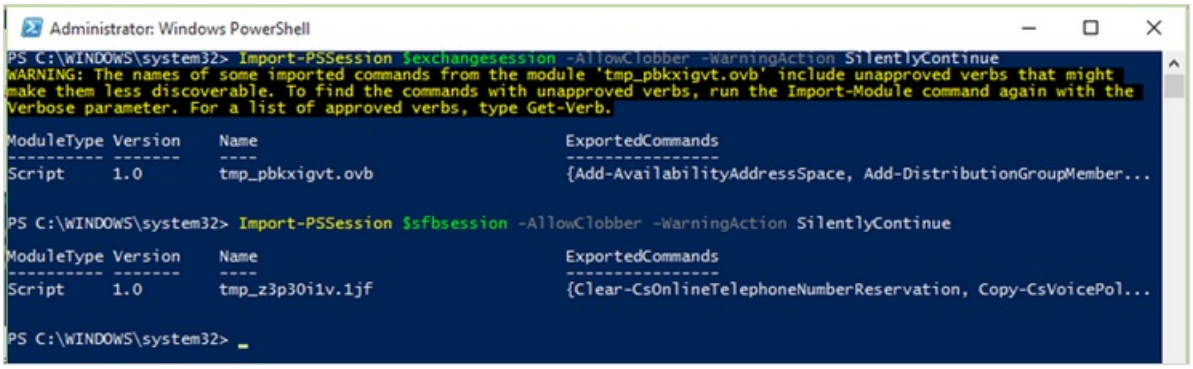
```
$exchangeSession = New-PSSession -ConfigurationName Microsoft.Exchange -ConnectionUri
"https://outlook.office365.com/powershell-liveid/" -Credential $cred -Authentication "Basic" –
AllowRedirection
```



6. Now you have to import the Skype for Business Online Session and the Exchange Online session you have just created, which will import the Exchange and Skype Commands so you can use them locally.

```
Import-PSSession $exchangesession -AllowClobber -WarningAction SilentlyContinue
Import-PSSession $sfbsession -AllowClobber -WarningAction SilentlyContinue
```

Note that this could take a while to complete.



7. Once you're connected to the online services you need to run a few more cmdlets to configure this account as a Surface Hub device account.

Use PowerShell to configure Exchange properties of the account

Now that you're connected to the online services, you can finish setting up the device account. You'll use the device account email address to:

- Change the mailbox type from regular to room.
- Set the password and enable the room mailbox account
- Change various Exchange properties
- Set the user account password to never expire.

1. You'll need to enter the account's mail address and create a variable with that value:

```
$mailbox = (Get-Mailbox <your device account's alias>)
```

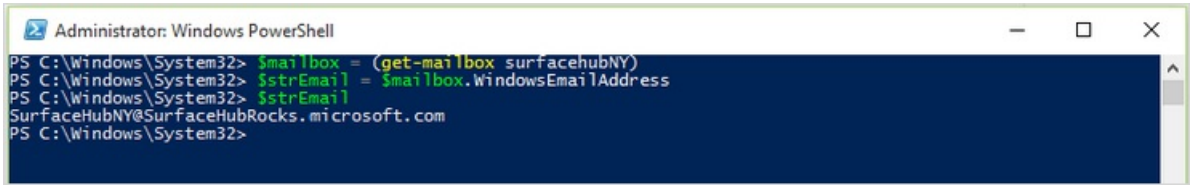
To store the value get it from the mailbox:

```
$strEmail = $mailbox.WindowsEmailAddress
```

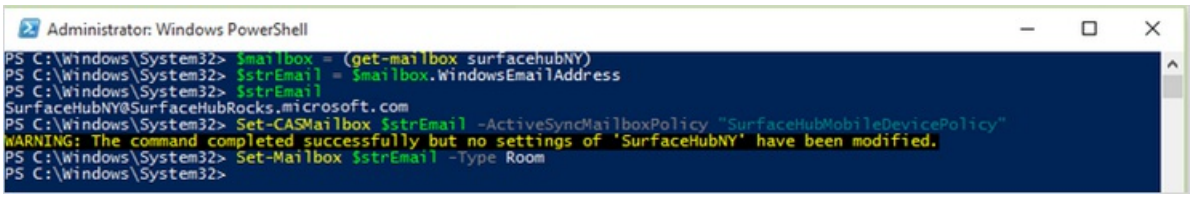
Print the value:

```
$strEmail
```

You will see the correct email address.



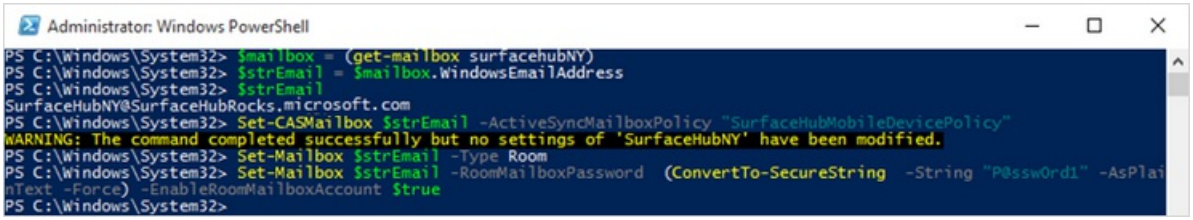
2. You need to convert the account into to a room mailbox, so run:




```
Set-Mailbox $strEmail -Type Room
```

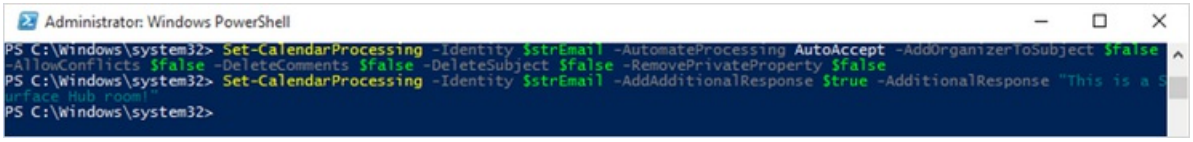
3. In order for the device account to be authenticated on a Surface Hub, you need to enable the room mailbox account and set a password, so the account can be used by the device to get meeting information using ActiveSync and log in to Skype for Business.

```
Set-Mailbox $strEmail -RoomMailboxPassword (ConvertTo-SecureString  -String "<your password>" -
AsPlainText -Force) -EnableRoomMailboxAccount $true
```



4. Various Exchange properties can be set on the device account to improve the meeting experience. You can see which properties need to be set in the Exchange properties section.

```
Set-CalendarProcessing -Identity $acctUpn -AutomateProcessing AutoAccept -AddOrganizerToSubject $false –
AllowConflicts   $false –DeleteComments $false -DeleteSubject $false -RemovePrivateProperty $false
Set-CalendarProcessing -Identity $acctUpn -AddAdditionalResponse $true -AdditionalResponse "This is a
Surface Hub room!"
```



5. If you decide to have the password not expire, you can set that with PowerShell cmdlets too. See Password management for more information.

```
Set-MsolUser -UserPrincipalName $strEmail -PasswordNeverExpires $True
```

Enable the account with Skype for Business

Enable the device account with Skype for Business.

In order to enable Skype for Business, your environment will need to meet the following prerequisites:

- You'll need to have Lync Online (Plan 2) or higher in your O365 plan. The plan needs to support conferencing capability.
- If you need Enterprise Voice (PSTN telephony) using telephony service providers for the Surface Hub, you need Lync Online (Plan 3).
- Your tenant users must have Exchange mailboxes.
- Your Surface Hub account does require a Lync Online (Plan 2) or Lync Online (Plan 3) license, but it does not require an Exchange Online license.

1. Start by creating a remote PowerShell session from a PC.

```
Import-Module LyncOnlineConnector
$cssess=New-CsOnlineSession -Credential $cred
Import-PSSession $cssess -AllowClobber
```

2. To enable your Surface Hub account for Skype for Business Server, run this cmdlet:

```
Enable-CsMeetingRoom -Identity $rm -RegistrarPool
"sippoolbl20a04.infra.lync.com" -SipAddressType EmailAddress
```

If you aren't sure what value to use for the `RegistrarPool` parameter in your environment, you can get the value from an existing Skype for Business user using this cmdlet:

```
Get-CsOnlineUser -Identity 'alice@contoso.microsoft.com'| fl *registrarpool*
```

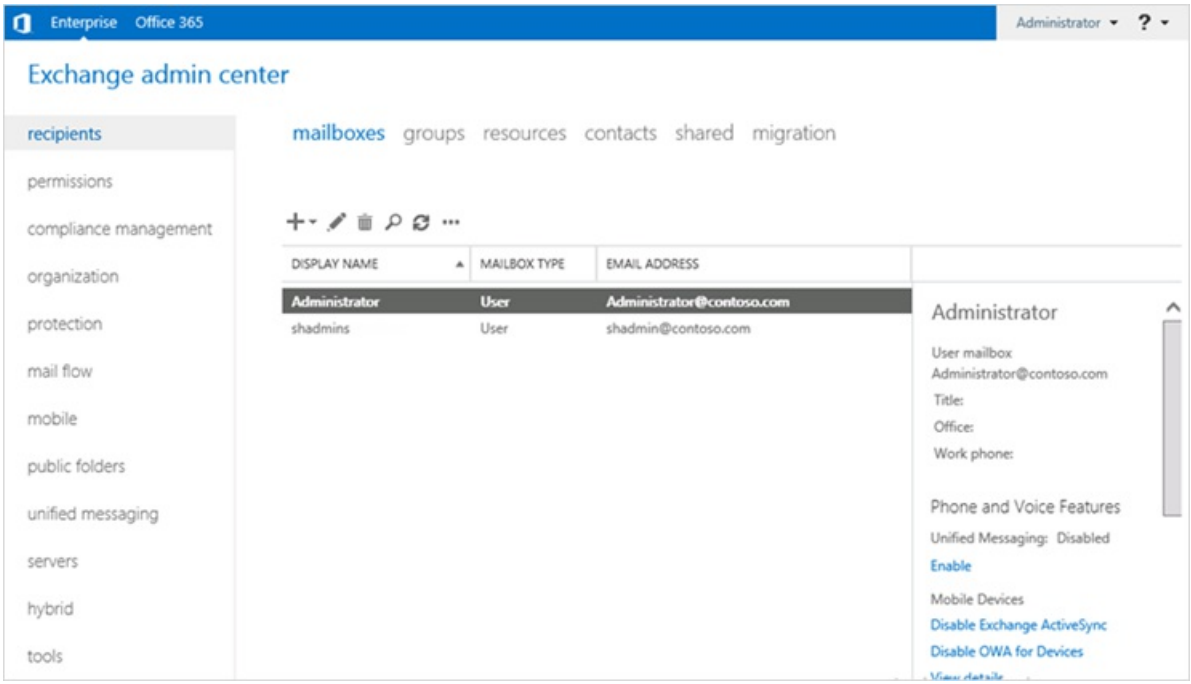
Create a device account using the Exchange Admin Center

You can use the Exchange Admin Center to create a device account:

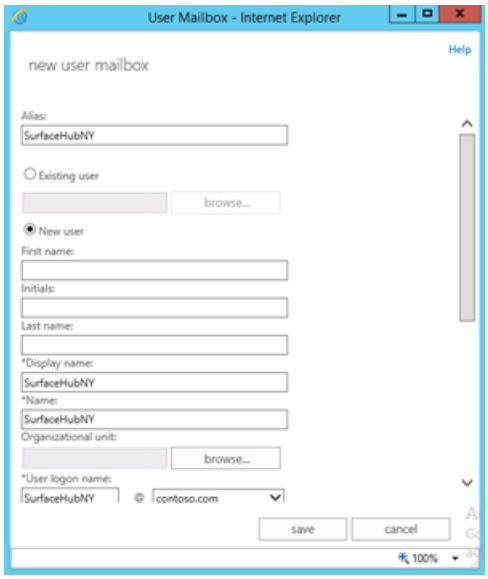
1. Create an account and mailbox with the Exchange Admin Center.
2. Create a mobile device mailbox policy from the Exchange Admin Center.
3. Use PowerShell to configure the account.
4. Enable the account with Skype for Business.

Create an account and mailbox with the Exchange Admin Center

1. Sign in to your Exchange Admin Center using Exchange admin credentials.
2. Once you are at the Exchange Admin Center (EAC), navigate to **Recipients** in the left panel.



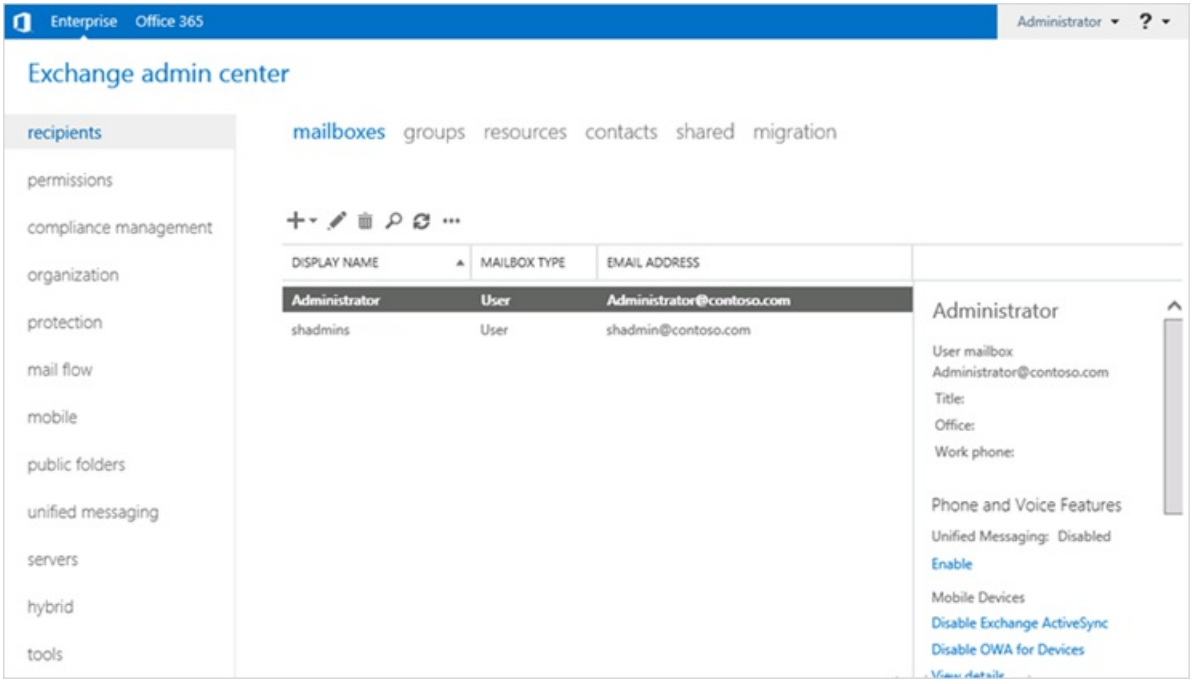
3. On the controls above the list of mailboxess, choose **+** to create a new one, and provide a **Display name**, **Name**, and **User logon name**, and then click **Save**.



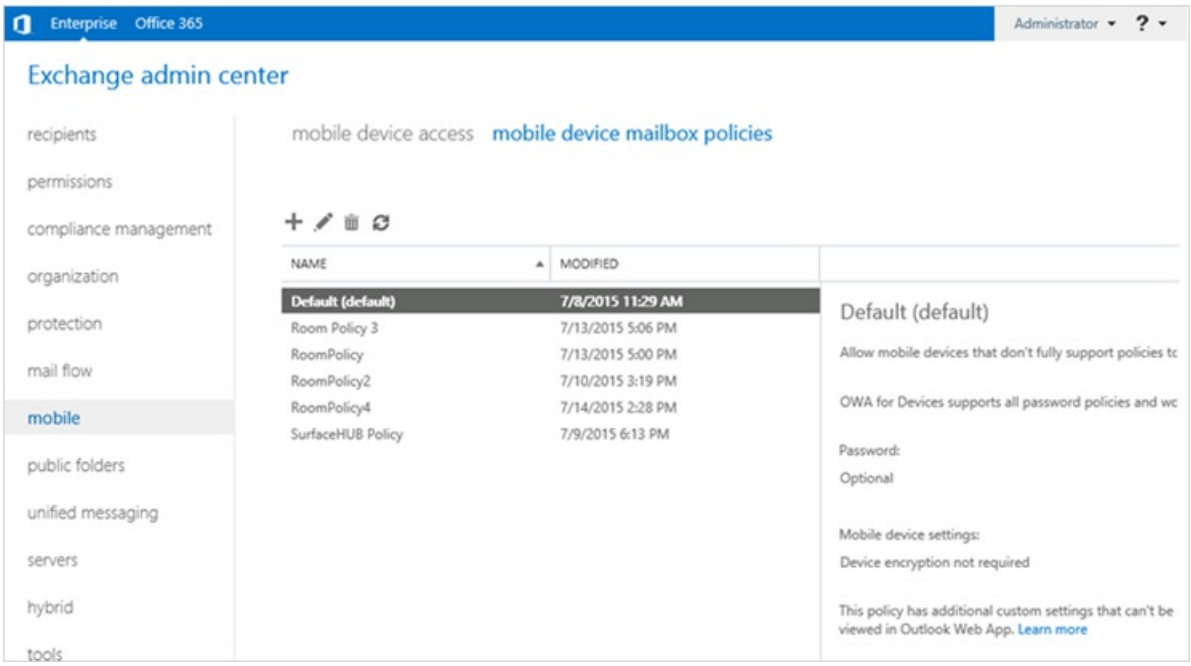
Create a mobile device mailbox policy from the Exchange Admin Center

> **Note** If you want to create and assign a policy to the account you created, and are using Exchange 2010, look up the corresponding information regarding policy creation and policy assignment when using the EMC (Exchange management console).

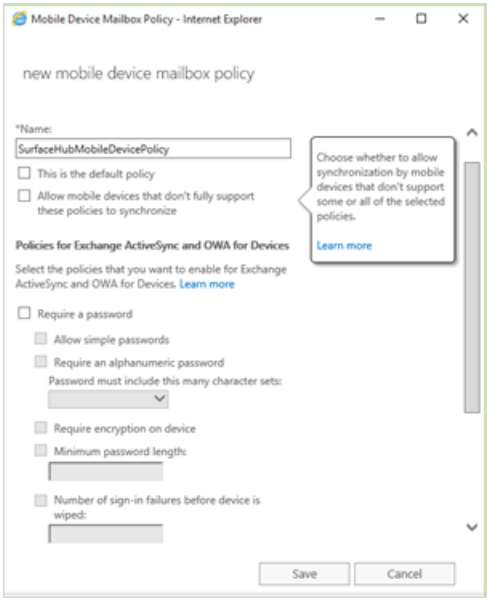
1. Go to the Exchange Admin Center.



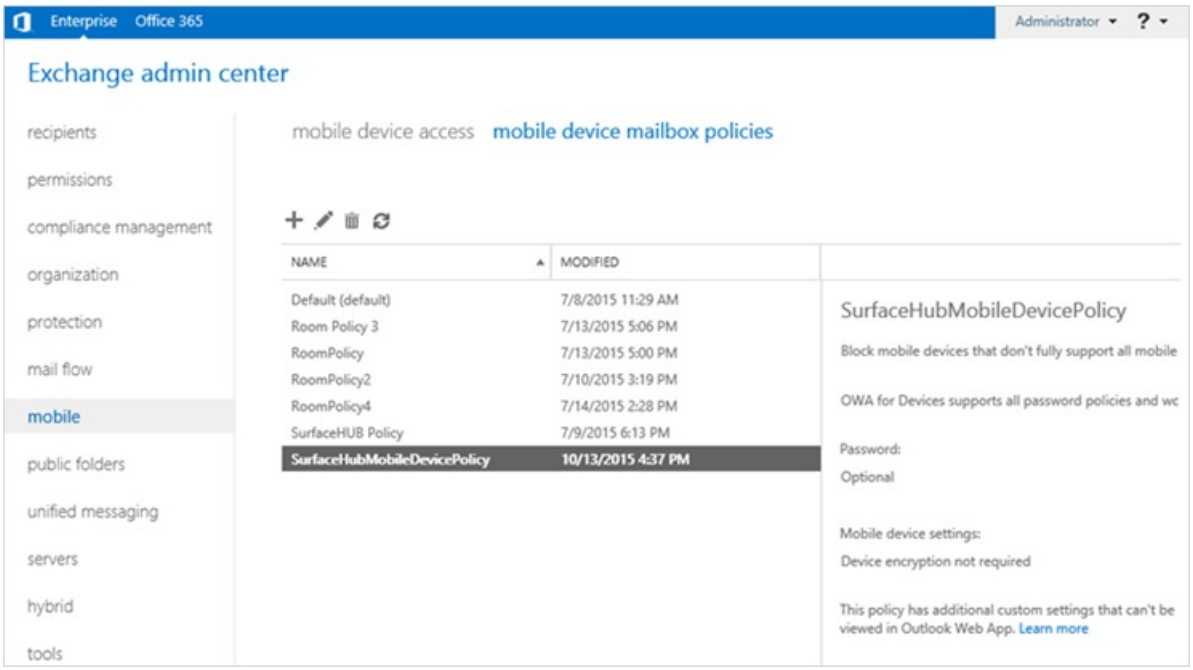
2. To create a mobile device mailbox policy, click **Mobile** from the left panel, then **Mobile device mailbox policies**. Surface Hubs require an account with a mobile device mailbox policy that does not require a password, so if you already have an existing policy that matches this requirement, you can apply that policy to the account. Otherwise use the following steps to create a new one to be used only for Surface Hub device accounts.



3. To create a new mobile device account mailbox policy, click the **+** button from the controls above the list of policies to add a new policy. For the name provide a name that will help you distinguish this policy from other device accounts (for example, *SurfaceHubDeviceMobilePolicy*). The policy must not be password-protected, so make sure **Require a Password** remains unchecked, then click **Save**.

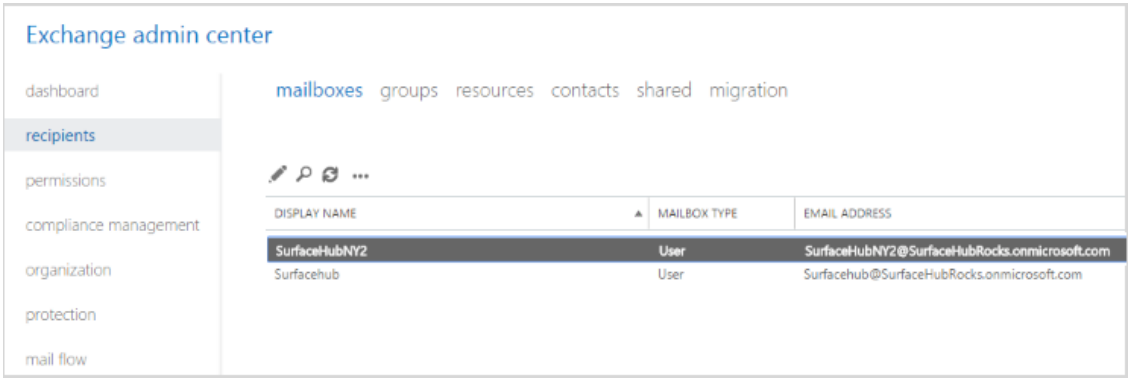


4. After you have created the new mobile device mailbox policy, go back to the Exchange Admin Center and you will see the new policy listed.

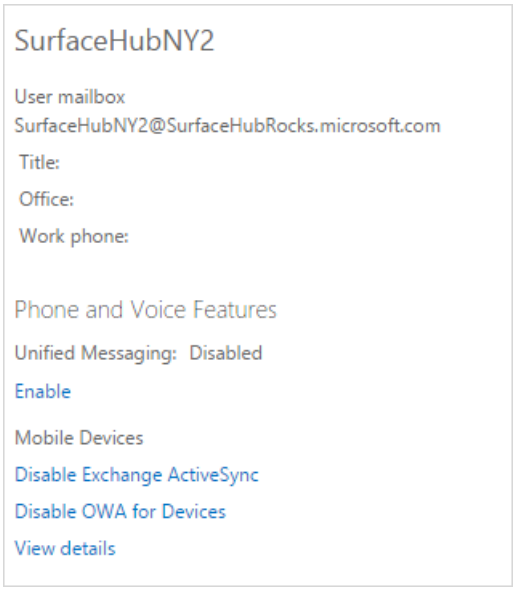


5. To apply the ActiveSync policy without using PowerShell, you can do the following:

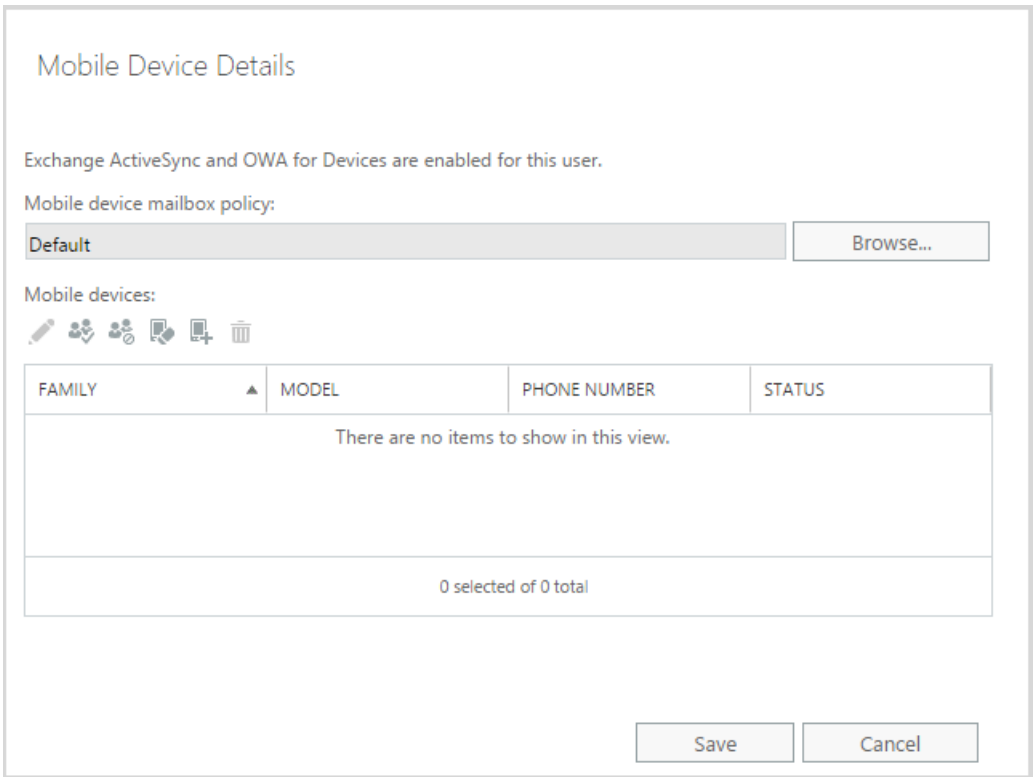
   - In the EAC, click **Recipients** > **Mailboxes** and select a mailbox.



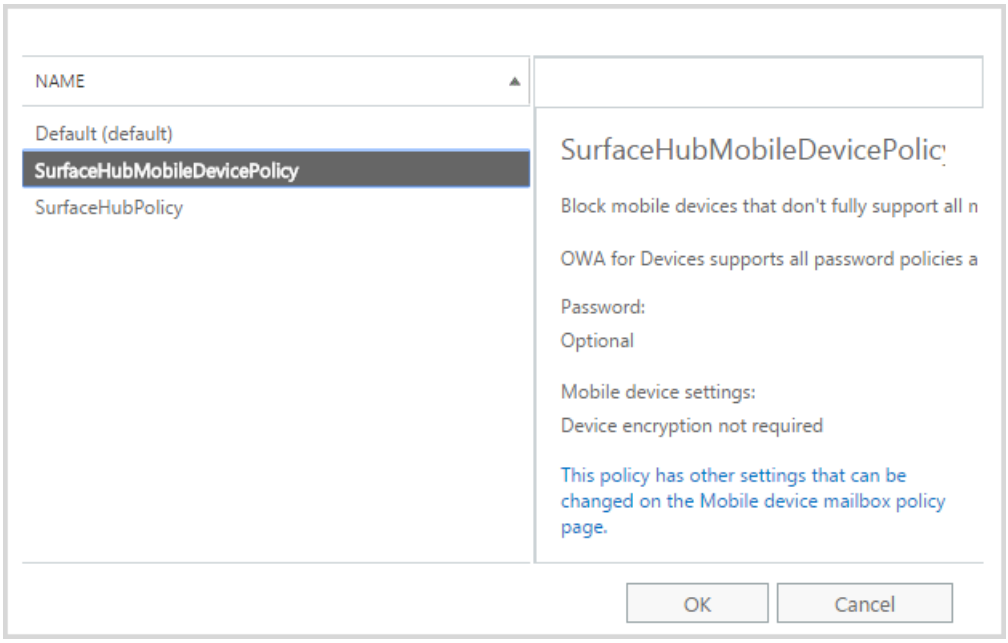
   - In the **Details** pane, scroll to **Phone and Voice Features** and click **View details** to display the **Mobile Device Details** screen.



   - The mobile device mailbox policy that's currently assigned is displayed. To change the mobile device mailbox policy, click **Browse**.



- Choose the appropriate mobile device mailbox policy from the list, click **OK** and then click **Save**.



Use PowerShell to configure the account

Now that you're connected to the online services, you can finish setting up the device account. You'll use the device account email address to:

- Change the mailbox type from regular to room.
- Change various Exchange properties
- Set the user account password to never expire.

1. You'll need to enter the account's mail address and create a variable with that value:

```
$mailbox = (Get-Mailbox <your device account's alias>)
```

To store the value got it from the mailbox:

```
$strEmail = $mailbox.WindowsEmailAddress
```

Print the value by running:

```
$strEmail
```

You will see the correct email address.

2. You need to convert the account into to a room mailbox, so run:

```
Set-Mailbox $strEmail -Type Room
```

3. In order for the device account to be authenticated on a Surface Hub, you need to enable the room mailbox account and set a password, so the account can be used by the device to get meeting information using ActiveSync and log in to Skype for Business.

```
Set-Mailbox $strEmail -RoomMailboxPassword (ConvertTo-SecureString  -String "<your password>" -
AsPlainText -Force) -EnableRoomMailboxAccount $true
```

4. Various Exchange properties can be set on the device account to improve the meeting experience. You can see which properties need to be set in the Exchange properties section.

```
Set-CalendarProcessing -Identity $acctUpn -AutomateProcessing AutoAccept -AddOrganizerToSubject $false –
AllowConflicts   $false –DeleteComments $false -DeleteSubject $false -RemovePrivateProperty $false
Set-CalendarProcessing -Identity $acctUpn -AddAdditionalResponse $true -AdditionalResponse "This is a
Surface Hub room!"
```

5. Now we have to set some properties in AD. To do that, you need the alias of the account (this is the part of the UPN that becomes before the "@").

```
$strAlias = "<your device account's alias>"
```

6. The user needs to be enabled in AD before it can authenticate with a Surface Hub. Run:

```
Set-ADUser $strAlias -Enabled $True
```

7. If you decide to have the password not expire, you can set that with PowerShell cmdlets too. See Password management for more information.

```
Set-ADUser $strAlias -PasswordNeverExpires $True
```

Enable the account with Skype for Business

Enable the device account with Skype for Business.

In order to enable Skype for Business, your environment will need to meet the following prerequisites:

- You'll need to have Lync Online (Plan 2) or higher in your O365 plan. The plan needs to support conferencing capability.
- If you need Enterprise Voice (PSTN telephony) using telephony service providers for the Surface Hub, you need Lync Online (Plan 3).

- Your tenant users must have Exchange mailboxes.
- Your Surface Hub account does require a Lync Online (Plan 2) or Lync Online (Plan 3) license, but it does not require an Exchange Online license.

1. Start by creating a remote PowerShell session from a PC.

   ```
   Import-Module LyncOnlineConnector
   $cssess=New-CsOnlineSession -Credential $cred
   Import-PSSession $cssess -AllowClobber
   ```

2. To enable your Surface Hub account for Skype for Business Server, run this cmdlet:

   ```
   Enable-CsMeetingRoom -Identity $rm -RegistrarPool
   "sippoolbl20a04.infra.lync.com" -SipAddressType EmailAddress
   ```

   If you aren't sure what value to use for the `RegistrarPool` parameter in your environment, you can get the value from an existing Skype for Business user using this cmdlet:

   ```
   Get-CsOnlineUser -Identity ‘alice@contoso.microsoft.com’| fl *registrarpool*
   ```

Microsoft Exchange properties (Surface Hub)

5/4/2017 • 1 min to read • Edit Online

Some Microsoft Exchange properties of the device account must be set to particular values to have the best meeting experience on Microsoft Surface Hub. The following table lists various Exchange properties based on PowerShell cmdlet parameters, their purpose, and the values they should be set to.

PROPERTY	DESCRIPTION	VALUE	IMPACT
AutomateProcessing	The AutomateProcessing parameter enables or disables calendar processing on the mailbox.	AutoAccept	The Surface Hub will be able to automatically accept or decline meeting requests based on its availability.
AddOrganizerToSubject	The AddOrganizerToSubject parameter specifies whether the meeting organizer's name is used as the subject of the meeting request.	$False	The welcome screen will not show the meeting organizer twice (instead of showing it as both the organizer and in the meeting subject).
AllowConflicts	The AllowConflicts parameter specifies whether to allow conflicting meeting requests.	$False	The Surface Hub will decline meeting requests that conflict with another meeting's time.
DeleteComments	The DeleteComments parameter specifies whether to remove or keep any text in the message body of incoming meeting requests.	$False	The message body of meetings can be retained and retrieved from a Surface Hub if you need it during a meeting.
DeleteSubject	The DeleteSubject parameter specifies whether to remove or keep the subject of incoming meeting requests.	$False	Meeting request subjects can be shown on the Surface Hub.
RemovePrivateProperty	The RemovePrivateProperty parameter specifies whether to clear the private flag for incoming meeting requests.	$False	Private meeting subjects will show as Private on the welcome screen.

PROPERTY	DESCRIPTION	VALUE	IMPACT
AddAdditionalResponse	The AddAdditionalResponse parameter specifies whether additional information will be sent from the resource mailbox when responding to meeting requests.	$True	When a response is sent to a meeting request, custom text will be provided in the response.
AdditionalResponse	The AdditionalResponse parameter specifies the additional information to be included in responses to meeting requests. **Note** This text will not be sent unless AddAdditionalResponse is set to $True.	Your choice—the additional response can be used to inform people how to use a Surface Hub or point them towards resources.	Adding an additional response message can provide people an introduction to how they can use a Surface Hub in their meeting.

Applying ActiveSync policies to device accounts (Surface Hub)

5/4/2017 • 1 min to read • Edit Online

The Microsoft Surface Hub's device account uses ActiveSync to sync mail and calendar. This allows people to join and start scheduled meetings from the Surface Hub, and allows them to email any whiteboards they have made during their meeting.

For these features to work, the ActiveSync policies for your organization must be configured as follows:

- There can't be any global policies that block synchronization of the resource mailbox that's being used by the Surface Hub's device account. If there is such a blocking policy, you need to whitelist the Surface Hub as an allowed device.
- You must set a mobile device mailbox policy where the **PasswordEnabled** setting is set to False. Other mobile device mailbox policy settings are not compatible with the Surface Hub.

Whitelisting the DeviceID

Your organization may have a global policy that prevents syncing of device accounts provisioned on Surface Hubs. To configure this property, see Allowing device IDs for ActiveSync.

Setting PasswordEnabled

The device account must have an ActiveSync policy where the **PasswordEnabled** attribute is set to False or 0. To configure this property, see Creating a Surface Hub-compatible Microsoft Exchange ActiveSync policy.

Password management (Surface Hub)

5/4/2017 • 1 min to read • Edit Online

Every Microsoft Surface Hub device account requires a password to authenticate and enable features on the device. For security reasons, you may want to change (or "rotate") this password regularly. However, if the device account's password changes, the password that was previously stored on the Surface Hub will be invalid, and all features that depend on the device account will be disabled. You will need to update the device account's password on the Surface Hub from the Settings app to re-enable these features.

To simplify password management for your Surface Hub device accounts, there are two options:

1. Turn off password expiration for the device account.
2. Allow the Surface Hub to automatically rotate the device account's password.

Turn off password rotation for the device account

Set the device account's **PasswordNeverExpires** property to True. You should verify whether this meets your organization's security requirements.

Allow the Surface Hub to automatically rotate the device account's password

The Surface Hub can manage a device account's password by changing it frequently without requiring you to manually update the device account's information. You can enable this feature in **Settings**. Once enabled, the device account's password will change weekly during maintenance hours.

Note that when the device account's password is changed, you will not be shown the new password. If you need to sign in to the account, or to provide the password again (for example, if you want to change the device account settings on the Surface Hub), then you'll need use Active Directory or the Office 365 admin portal to reset the password.

> **IMPORTANT**
>
> If your organization uses a hybrid topology (some services are hosted on-premises and some are hosted online through Office 365), you must setup the device account in **domain\username** format. Otherwise, password rotation will not work.

Create provisioning packages (Surface Hub)

5/4/2017 • 8 min to read • Edit Online

This topic explains how to create a provisioning package using the Windows Imaging and Configuration Designer (ICD), and apply it to Surface Hub devices. For Surface Hub, you can use provisioning packages to add certificates, install Universal Windows Platform (UWP) apps, and customize policies and settings.

You can apply a provisioning package using a USB during first run, or through the **Settings** app.

Advantages

- Quickly configure devices without using a MDM provider.
- No network connectivity required.
- Simple to apply.

Learn more about the benefits and uses of provisioning packages.

Requirements

To create and apply a provisioning package to a Surface Hub, you'll need the following:

- Windows Imaging and Configuration Designer (ICD), which is installed as a part of the Windows 10 Assessment and Deployment Kit (ADK).
- A PC running Windows 10.
- A USB flash drive.
- If you apply the package using the **Settings** app, you'll need device admin credentials.

You'll create the provisioning package on a PC running Windows 10, save the package to a USB drive, and then deploy it to your Surface Hub.

Supported items for Surface Hub provisioning packages

Currently, you can add these items to provisioning packages for Surface Hub:

- **Certificates** - You can add certificates, if needed, to authenticate to Microsoft Exchange.
- **Universal Windows Platform (UWP) apps** - You can install UWP apps. This can be an offline-licensed app from the Microsoft Store for Business, or an app created by an in-house dev.
- **Policies** - Surface Hub supports a subset of the policies in the Policy configuration service provider. Some of those policies can be configured with ICD.
- **Settings** - You can configure any setting in the SurfaceHub configuration service provider.

Create the provisioning package

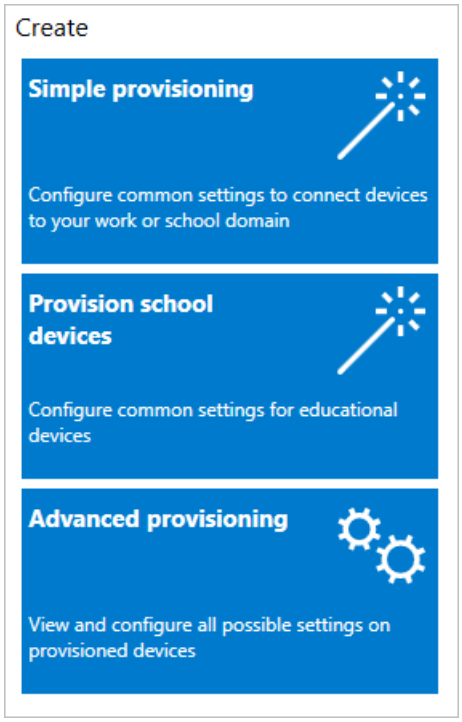
Use the Windows Imaging and Configuration Designer (ICD) tool included in the Windows Assessment and Deployment Kit (ADK) for Windows 10 to create a provisioning package. When you install the ADK, you can choose to install only the Imaging and Configuration Designer (ICD). Install the ADK.

1. Open Windows ICD (by default,

```
%windir%\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Imaging and Configuration Designer\x86\ICD.exe
```

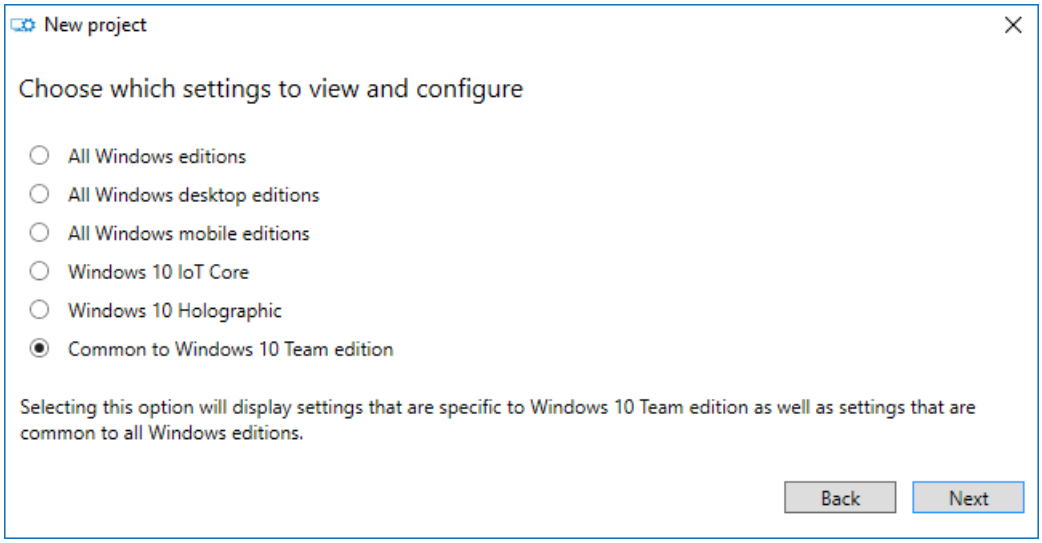
).

2. Click **Advanced provisioning**.

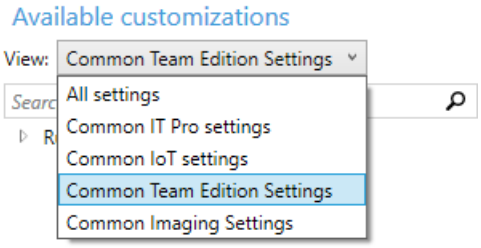


3. Name your project and click **Next**.

4. Select **Common to Windows 10 Team edition**, click **Next**, and then click **Finish**.



5. In the project, under **Available customizations**, select **Common Team edition settings**.



Add a certificate to your package

You can use provisioning packages to install certificates that will allow the device to authenticate to Microsoft Exchange.

> **NOTE**
> Provisioning packages can only install certificates to the device (local machine) store, and not to the user store. If your organization requires that certificates must be installed to the user store, use Mobile Device Management (MDM) to deploy these certificates. See your MDM solution documentation for details.

1. In the **Available customizations** pane, go to **Runtime settings** > **Certificates** > **ClientCertificates**.
2. Enter a **CertificateName** and then click **Add**.
3. Enter the **CertificatePassword**.
4. For **CertificatePath**, browse and select the certificate.
5. Set **ExportCertificate** to **False**.
6. For **KeyLocation**, select **Software only**.

Add a Universal Windows Platform (UWP) app to your package

Before adding a UWP app to a provisioning package, you need the app package (either an .appx, or .appxbundle) and any dependency files. If you acquired the app from the Microsoft Store for Business, you will also need the *unencoded* app license. See Distribute offline apps to learn how to download these items from the Microsoft Store for Business.

1. In the **Available customizations** pane, go to **Runtime settings** > **UniversalAppInstall** > **DeviceContextApp**.
2. Enter a **PackageFamilyName** for the app and then click **Add**. For consistency, use the app's package family name. If you acquired the app from the Microsoft Store for Business, you can find the package family name in the app license. Open the license file using a text editor, and use the value between the <PFM>... </PFM> tags.
3. For **ApplicationFile**, click **Browse** to find and select the target app (either an *.appx or *.appxbundle).
4. For **DependencyAppxFiles**, click **Browse** to find and add any dependencies for the app. For Surface Hub, you will only need the x64 versions of these dependencies.

If you acquired the app from the Microsoft Store for Business, you will also need to add the app license to your provisioning package.

1. Make a copy of the app license, and rename it to use a **.ms-windows-store-license** extension. For example, "example.xml" becomes "example.ms-windows-store-license".
2. In ICD, in the **Available customizations** pane, go to **Runtime settings** > **UniversalAppInstall** > **DeviceContextAppLicense**.
3. Enter a **LicenseProductId** and then click **Add**. For consistency, use the app's license ID from the app license. Open the license file using a text editor. Then, in the <License> tag, use the value in the **LicenseID** attribute.
4. Select the new **LicenseProductId** node. For **LicenseInstall**, click **Browse** to find and select the license file that you renamed in Step 1.

Add a policy to your package

Surface Hub supports a subset of the policies in the Policy configuration service provider. Some of those policies can be configured with ICD.

1. In the **Available customizations** pane, go to **Runtime settings** > **Policies**.

2. Select one of the available policy areas.
3. Select and set the policy you want to add to your provisioning package.

Add Surface Hub settings to your package

You can add settings from the SurfaceHub configuration service provider to your provisioning package.

1. In the **Available customizations** pane, go to **Runtime settings** > **WindowsTeamSettings**.
2. Select one of the available setting areas.
3. Select and set the setting you want to add to your provisioning package.

Build your package

1. When you are done configuring the provisioning package, on the **File** menu, click **Save**.
2. Read the warning that project files may contain sensitive information, and click **OK**.

> **IMPORTANT**
> When you build a provisioning package, you may include sensitive information in the project files and in the provisioning package (.ppkg) file. Although you have the option to encrypt the .ppkg file, project files are not encrypted. You should store the project files in a secure location and delete the project files when they are no longer needed.

3. On the **Export** menu, click **Provisioning package**.
4. Change **Owner** to **IT Admin**, which will set the precedence of this provisioning package higher than provisioning packages applied to this device from other sources.
5. Set a value for **Package Version**, and then select **Next.**

> **TIP**
> You can make changes to existing packages and change the version number to update previously applied packages.

6. Optional: You can choose to encrypt the package and enable package signing.
   - **Enable package encryption** - If you select this option, an auto-generated password will be shown on the screen.
   - **Enable package signing** - If you select this option, you must select a valid certificate to use for signing the package. You can specify the certificate by clicking **Browse...** and choosing the certificate you want to use to sign the package.

     > **IMPORTANT**
     > We recommend that you include a trusted provisioning certificate in your provisioning package. When the package is applied to a device, the certificate is added to the system store and any package signed with that certificate thereafter can be applied silently.

7. Click **Next** to specify the output location where you want the provisioning package to go once it's built. By default, Windows ICD uses the project folder as the output location.

   Optionally, you can click **Browse** to change the default output location.

8. Click **Next**.
9. Click **Build** to start building the package. The project information is displayed in the build page and the progress bar indicates the build status.

   If you need to cancel the build, click **Cancel**. This cancels the current build process, closes the wizard, and takes you back to the **Customizations Page**.
10. If your build fails, an error message will show up that includes a link to the project folder. You can scan the logs to determine what caused the error. Once you fix the issue, try building the package again.

    If your build is successful, the name of the provisioning package, output directory, and project directory will be shown.

    - If you choose, you can build the provisioning package again and pick a different path for the output package. To do this, click **Back** to change the output package name and path, and then click **Next** to start another build.
    - If you are done, click **Finish** to close the wizard and go back to the **Customizations Page**.
11. Select the **output location** link to go to the location of the package. Copy the .ppkg to an empty USB flash drive.

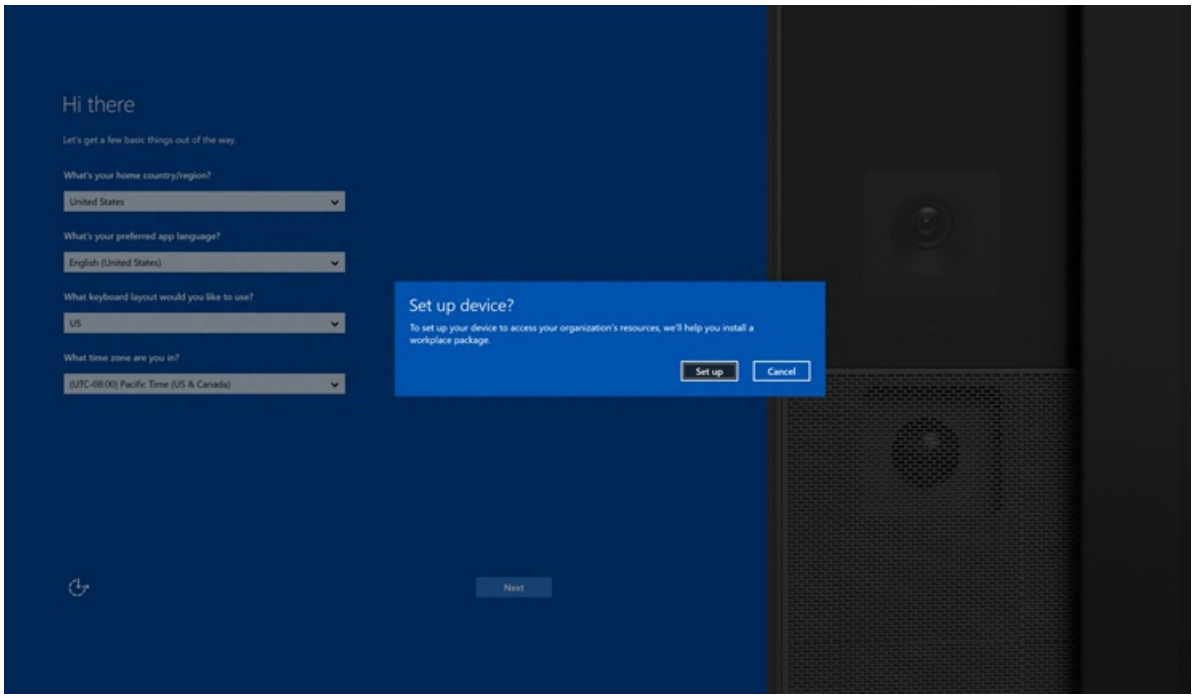
Apply a provisioning package to Surface Hub

There are two options for deploying provisioning packages to a Surface Hub. You can apply a provisioning packing during the first run wizard, or using Settings.

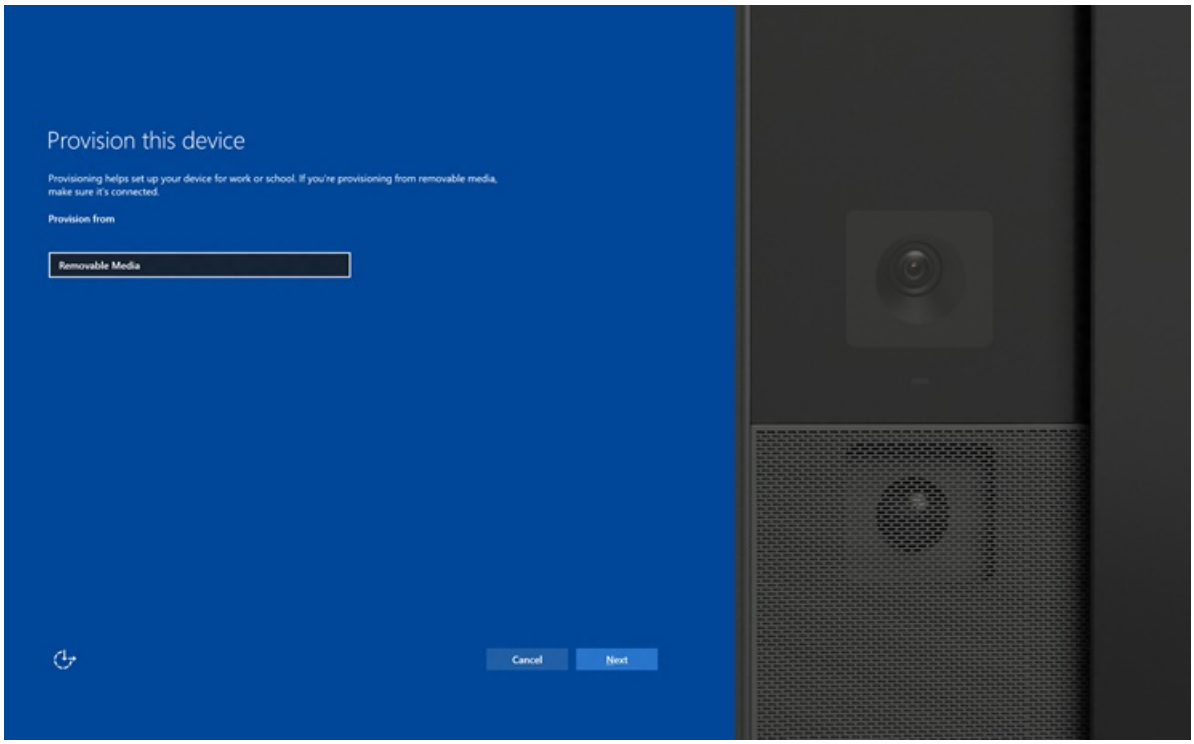
Apply a provisioning package during first run

> **IMPORTANT**
> Only use provisioning packages to install certificates during first run. Use the **Settings** app to install apps and apply other settings.

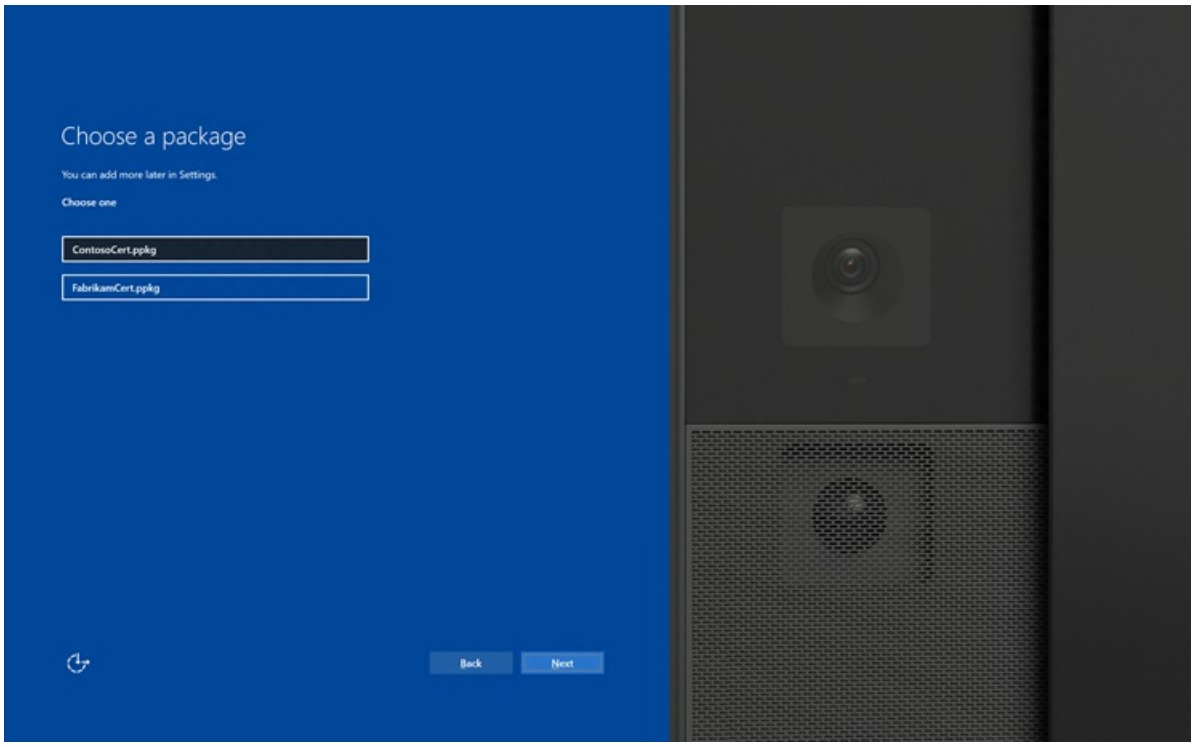
1. When you turn on the Surface Hub for the first time, the first-run program will display the **Hi there page**. Make sure that the settings are properly configured before proceeding.
2. Insert the USB flash drive containing the .ppkg file into the Surface Hub. If the package is in the root directory of the drive, the first-run program will recognize it and ask if you want to set up the device. Select **Set up**.



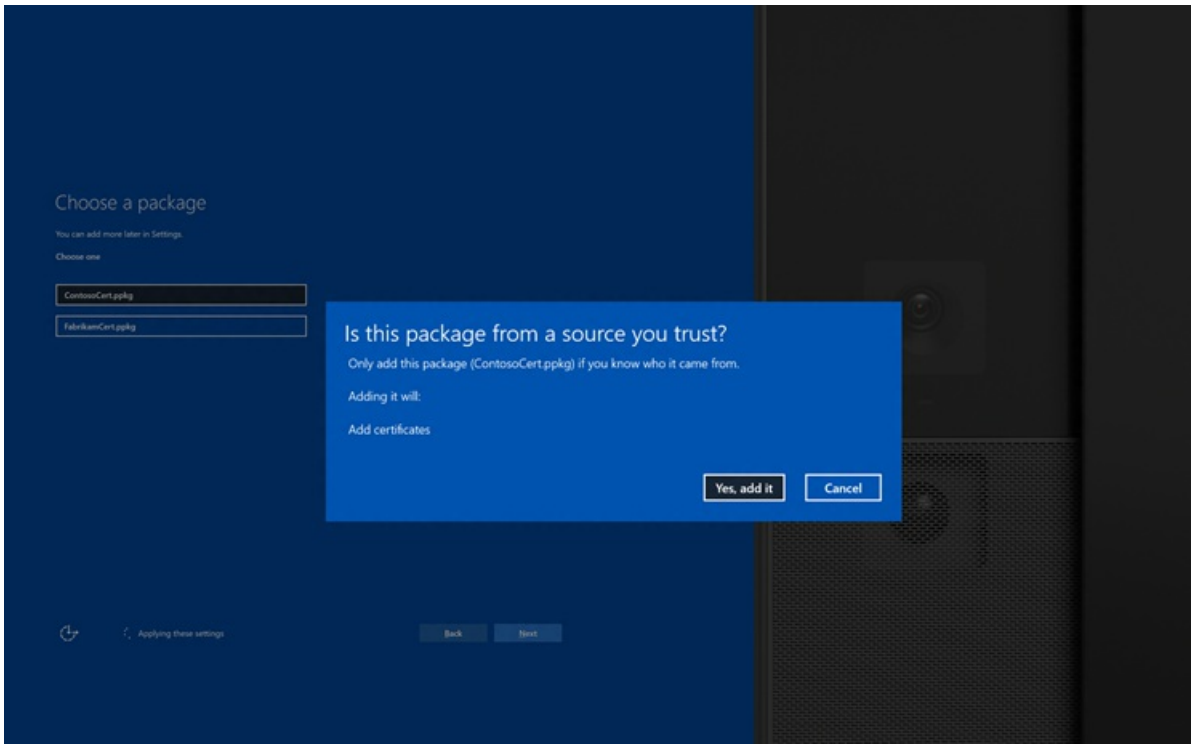
3. The next screen asks you to select a provisioning source. Select **Removable Media** and tap **Next**.



4. Select the provisioning package (*.ppkg) that you want to apply, and tap **Next**. Note that you can only install one package during first run.



5. The first-run program will show you a summary of the changes that the provisioning package will apply. Select **Yes, add it**. The package will be applied, and you'll be taken to the next page in the first-run program.



Apply a package using Settings

1. Insert the USB flash drive containing the .ppkg file into the Surface Hub.
2. From the Surface Hub, start **Settings** and enter the admin credentials when prompted.
3. Navigate to **This device** > **Device management**. Under **Provisioning packages**, select **Add or remove a provisioning package**.
4. Select **Add a package**.
5. Choose your provisioning package and select **Add**. You may have to re-enter the admin credentials if

prompted.

6. You'll see a summary of the changes that the provisioning package will apply. Select **Yes, add it**.

Admin group management (Surface Hub)

5/4/2017 • 4 min to read • Edit Online

Every Surface Hub can be configured locally using the Settings app on the device. To prevent unauthorized users from changing settings, the Settings app requires admin credentials to open the app.

Admin Group Management

You can set up administrator accounts for the device in one of three ways:

- Create a local admin account
- Domain join the device to Active Directory (AD)
- Azure Active Directory (Azure AD) join the device

Create a local admin account

To create a local admin, choose to use a local admin during first run. This will create a single local admin account on the Surface Hub with the username and password of your choice. Use these credentials to open the Settings app.

Note that the local admin account information is not backed by any directory service. We recommend you only choose a local admin if the device does not have access to Active Directory (AD) or Azure Active Directory (Azure AD). If you decide to change the local admin's password, you can do so in Settings. However, if you want to change from using the local admin account to using a group from your domain or Azure AD tenant, then you'll need to reset the device and go through the first-time program again.

Domain join the device to Active Directory (AD)

You can domain join the Surface Hub to your AD domain to allow users from a specified security group to configure settings. During first run, choose to use Active Directory Domain Services. You'll need to provide credentials that are capable of joining the domain of your choice, and the name of an existing security group. Anyone who is a member of that security group can enter their credentials and unlock Settings.

What happens when you domain join your Surface Hub?

Surface Hubs use domain join to:

- Grant admin rights to members of a specified security group in AD.
- Backup the device's BitLocker recovery key by storing it under the computer object in AD. See Save your BitLocker key for details.
- Synchronize the system clock with the domain controller for encrypted communication

Surface Hub does not support applying group policies or certificates from the domain controller.

> **NOTE**
> If your Surface Hub loses trust with the domain (for example, if you remove the Surface Hub from the domain after it is domain joined), you won't be able to authenticate into the device and open up Settings. If you decide to remove the trust relationship of the Surface Hub with your domain, reset the device first.

Azure Active Directory (Azure AD) join the device

You can Azure AD join the Surface Hub to allow IT pros from your Azure AD tenant to configure settings. During first run, choose to use Microsoft Azure Active Directory. You will need to provide credentials that are capable of joining the Azure AD tenant of your choice. After you successfully Azure AD join, the appropriate people will be

granted admin rights on the device.

By default, all **global administrators** will be given admin rights on an Azure AD joined Surface Hub. With **Azure AD Premium** or **Enterprise Mobility Suite (EMS)**, you can add additional administrators:

1. In the Azure classic portal, click **Active Directory**, and then click the name of your organization's directory.
2. On the **Configure** page, under **Devices** > **Additional administrators on Azure AD joined devices**, click **Selected**.
3. Click **Add**, and select the users you want to add as administrators on your Surface Hub and other Azure AD joined devices.
4. When you have finished, click the checkmark button to save your change.

What happens when you Azure AD join your Surface Hub?

Surface Hubs use Azure AD join to:

- Grant admin rights to the appropriate users in your Azure AD tenant.
- Backup the device's BitLocker recovery key by storing it under the account that was used to Azure AD join the device. See Save your BitLocker key for details.

> **IMPORTANT**
> Surface Hub does not currently support automatic enrollment to Microsoft Intune through Azure AD join. If your organization automatically enrolls Azure AD joined devices into Intune, you must disable this policy for Surface Hub before joining the device to Azure AD.

Which should I choose?

If your organization is using AD or Azure AD, we recommend you either domain join or Azure AD join, primarily for security reasons. People will be able to authenticate and unlock Settings with their own credentials, and can be moved in or out of the security groups associated with your domain.

OPTION	REQUIREMENTS	WHICH CREDENTIALS CAN BE USED TO ACCESS THE SETTINGS APP?
Create a local admin account	None	The user name and password specified during first run
Domain join to Active Directory (AD)	Your organization uses AD	Any AD user from a specific security group in your domain
Azure Active Directory (Azure AD) join the device	Your organization uses Azure AD Basic	Global administrators only
	Your organization uses Azure AD Premium or Enterprise Mobility Suite (EMS)	Global administrators and additional administrators

Set up Microsoft Surface Hub

5/4/2017 • 1 min to read • Edit Online

Set up instructions for Surface Hub include a setup worksheet, and a walkthrough of the first-run program.

Before you turn on your Microsoft Surface Hub for the first time, make sure you've completed the checklist at the end of the Prepare your environment for Surface Hub section, and that you have the information listed in the Setup worksheet. When you do power it on, the device will walk you through a series of setup screens. If you haven't properly set up your environment, or don't have the required information, you'll have to do extra work afterward making sure the settings are correct.

In this section

TOPIC	DESCRIPTION
Setup worksheet	When you've finished pre-setup and are ready to start first-time setup for your Surface Hub, make sure you have all the information listed in this section.
First-run program	The term "first run" refers to the series of steps you'll go through the first time you power up your Surface Hub, and means the same thing as "out-of-box experience" (OOBE). This section will walk you through the process.

Setup worksheet (Surface Hub)

5/4/2017 • 3 min to read • Edit Online

When you've finished pre-setup and are ready to start first-time setup for your Microsoft Surface Hub, make sure you have all the information listed in this section.

You should fill out one list for each Surface Hub you need to configure, although some information can be used on all Surface Hubs, like the proxy information or domain credentials. Some of this information may not be needed, depending on how you've decided to configure your device, or depending on how the environment is configured for your organization's infrastructure.

PROPERTY	WHAT THIS IS USED FOR	EXAMPLE	ACTUAL VALUE
Proxy information	If your network uses a proxy for network and/or Internet access, you must provide a script or server/port information.	Proxy script: `http://contoso/proxy.pa` - OR - Server and port info: 10.10.10.100, port 80	
Wireless network credentials (username and password)	If you decide to connect your device to Wi-Fi, and your wireless network requires user credentials.	admin1@contoso.com, #MyPassw0rd	
Device account UPN or Domain\username and device account password	This is the User Principal Name (UPN) or the domain\username, and the password of the device account. Mail, calendar, and Skype for Business depend on a compatible device account.	UPN: ConfRoom15@contoso.com, #Passw0rd1 - OR - Domain and username: CONTOSO\ConfRoom15, #Passw0rd1	
Device account Microsoft Exchange server	This is the device account's Exchange server. Mail, calendar, and Skype for Business depend on a compatible device account. For mail and calendar to work, the device account must have a valid Exchange server. The device will try to find this automatically.	outlook.office365.com	

Device account Session Initiation Protocol (SIP) address	This is the device account's Skype for Business SIP address. Mail, calendar, and Skype for Business depend on a compatible device account. For Skype for Business to work, the device account must have a valid SIP address. The device will try to find this automatically.	sip: ConfRoom15@contoso.com	
Friendly name	The friendly name of the device is the broadcast name that people will see when they try to wirelessly connect to the Surface Hub. This name will be displayed prominently on the Surface Hub's screen. We suggest that the friendly name you choose is recognizable and unique so that people can distinguish one Surface Hub from another when trying to connect.	Conference Room 15	
Device name	The device name is the name that will be used for domain join, and is the identity you will see in your MDM provider if the device is enrolled into MDM. The device name you choose must not be the same name as any other device on the user's Active Directory domain (if you decide to domain join the device). The device cannot join the domain if its name is not unique.	confroom15	
IF YOU'RE JOINING AZURE AD			

Azure AD tenant user credentials (username and password)	If you decide to have people in your Azure Active Directory (Azure AD) organization become admins on the device, then you'll need to join Azure AD. To join Azure AD, you will need valid user credentials.	admin1@contoso.com, #MyPassw0rd	
IF YOU'RE JOINING A DOMAIN			
Domain to join	This is the domain you will need to join so that a security group of your choice can be admins for the device. You may need the fully qualified domain name (FQDN).	contoso (short name) OR contoso.corp.com (FQDN)	
Domain account credentials (username and password)	A domain can't be joined unless you provide sufficient account credentials to join the domain. Once you provide a domain to join and credentials to join the domain, then a security group of your choice can change settings on the device.	admin1, #MyPassw0rd	
Admin security group alias	This is a security group in your Active Directory (AD); any members of this security group can change settings on the device.	SurfaceHubAdmins	
IF YOU'RE USING A LOCAL ADMIN			
Local admin account credentials (username and password)	If you decide not to join an AD domain or Azure AD, you can create a local admin account on the device.	admin1, #MyPassw0rd	
IF YOU NEED TO INSTALL CERTIFICATES OR APPS			

USB drive	If you know before first run that you want to install certificates or universal apps, follow the steps in Create provisioning packages. Your provisioning packages will be created on a USB drive.		

First-run program (Surface Hub)

5/4/2017 • 17 min to read • Edit Online

The term "first run" refers to the series of steps you'll go through the first time you power up your Microsoft Surface Hub, and means the same thing as "out-of-box experience" (OOBE). This section will walk you through the process.

By now, you should have gone through all of the previous steps:

- Prepare your environment for Surface Hub
- Physically install your Surface Hub device, and
- Setup worksheet

Assuming that's the case, first run should be both simple and quick. The normal procedure goes through six steps:

1. Hi there page
2. Set up for you page
3. Device account page
4. Name this device page
5. Set up admins for this device page
6. Update the Surface Hub

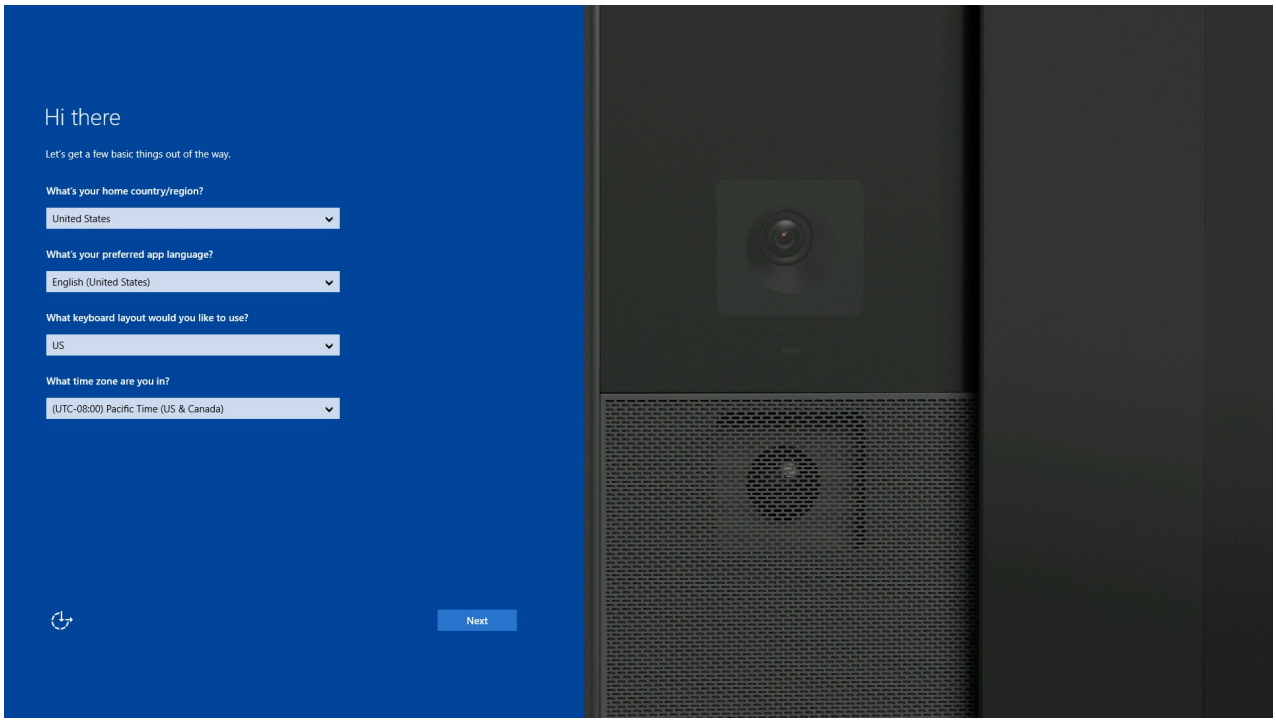
Each of these sections also contains information about paths you might take when something is different. For example, most Surface Hubs will use a wired network connection, but some of them will be set up with wireless instead. Details are described where appropriate.

> **Note** You should have the separate keyboard that came with your Surface Hub set up and ready before beginning. See the Surface Hub Setup Guide for details.

Hi there page

This is the first screen you'll see when you power up the Surface Hub for the first time. It's where you input localization information for your device.

> **Note** This is also where you begin the optional process of deploying a provisioning package. See Create provisioning packages if that's what you're doing.



Details

If the default values shown are correct, then you can click **Next** to go on. Otherwise, you'll need to enter data in the appropriate boxes.

- **Country/region:** Select the country or region where the Surface Hub will be used.
- **App language:** Apps and features will display in this language and language format.
- **Keyboard layout:** Select the keyboard layout for the on-screen and physical keyboards that will be used with your device.
- **Time zone:** Select the time zone where the Surface Hub will be used.

What happens?

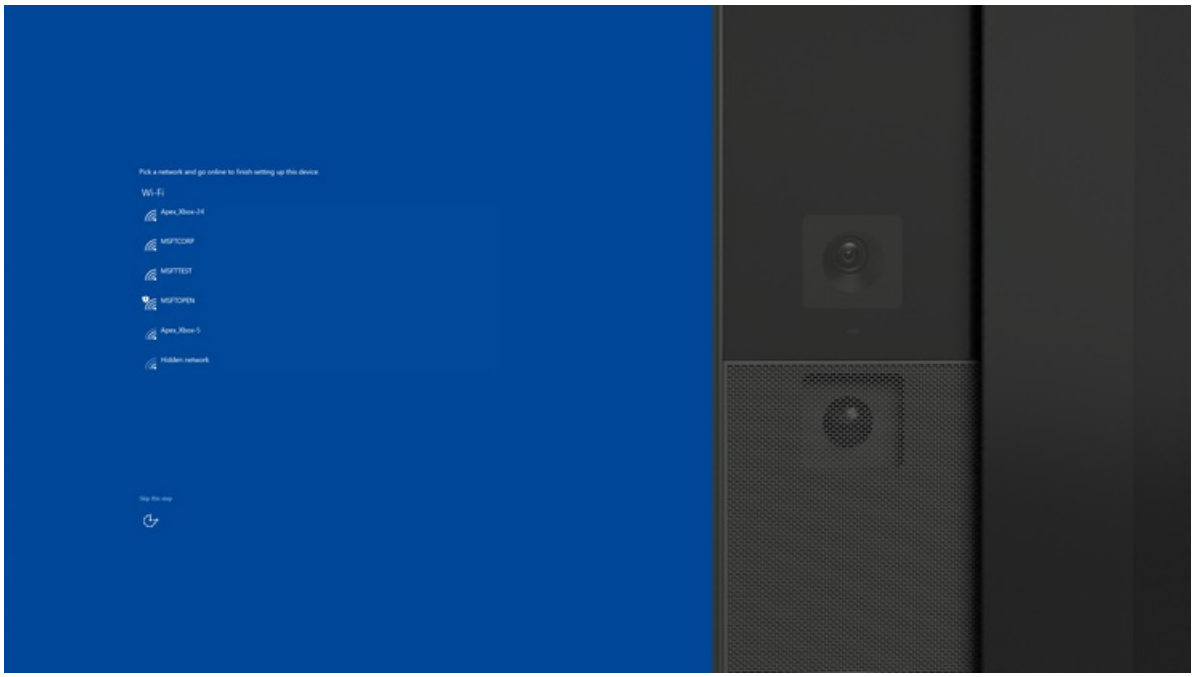
> **NOTE**
> Once the settings on this page are entered, you can't come back to this screen unless you reset the device (see Device reset). Make sure that the settings are properly configured before proceeding.

When the settings are accepted, the device will check for a wired network connection. If the connection is fine, it will display the Set up for you page. If there is a problem with the wired connection, the device will display the Network setup page.

If no wired connection can be found, then the device will attempt to set up a wireless connection, and will display the Network setup page.

Network setup page

If your device does not detect a wired connection that it can use to connect to a network or the Internet, you will see this page. Here you can either connect to a wireless network, or skip making the network connection.



Details

This screen is shown only if the device fails to detect a wired network. If you see this screen, you have three choices:

- You can select one of the wireless networks shown. If the network is secured, you'll be taken to a login page. See Wireless network setup for details.
- Click **Skip this step** to skip connecting to a network. You'll be taken to the Set up for you page. **Note** If you skip this, the device will not have a network connection, and nothing that requires a network connection will work on your Surface Hub, including system updates and email and calendar synchronization. You can connect to a wireless network later using Settings (see Wireless network managment).
- You can plug in a network cable while this screen is visible. The device will detect it, and will add **Next** to the screen. Click **Next** to continue with making the wired connection.

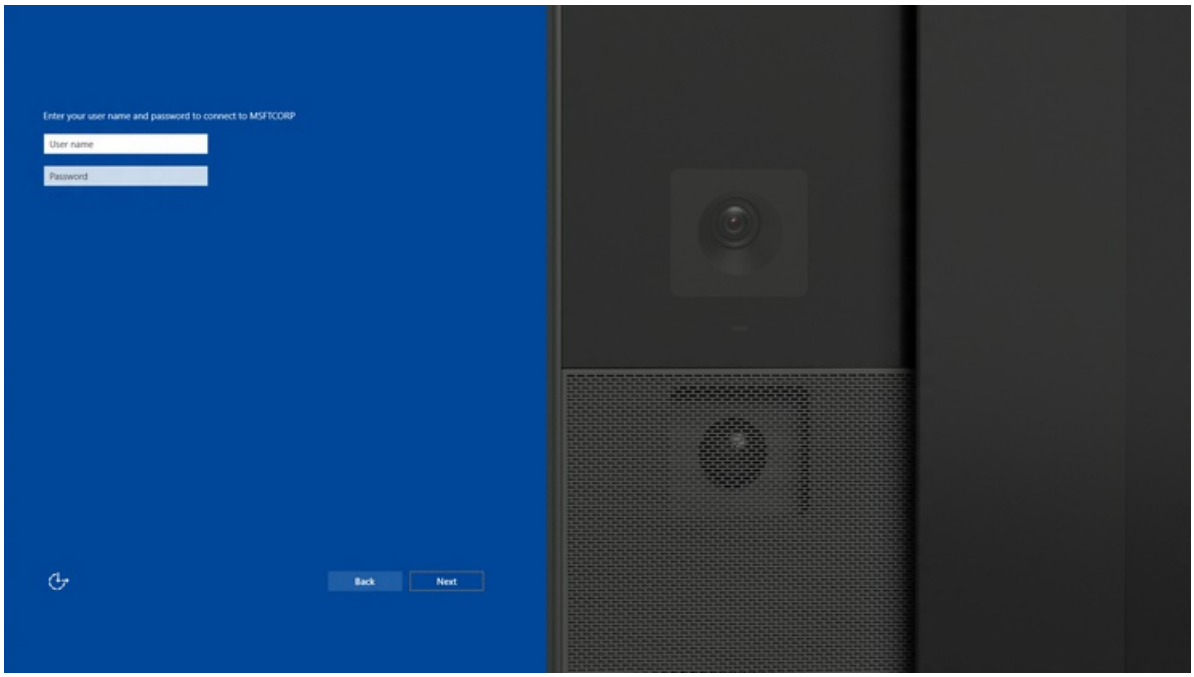
What happens?

If the device has a wired connection when it starts, and can establish a network or Internet connection, then this page will not be displayed. If you want to connect the device to a wireless connection, make sure no Ethernet cable is plugged in at first run, which will bring you to this screen. No matter what you choose to set up now, you can use Settings to set up different connections later.

If you want to connect to a secured wireless network from this page, click on the network of your choice, and then provide the necessary information (password or account credentials) to connect. See Wireless network setup.

Wireless network setup

This page will be shown when you've selected a secured wireless network.



Details

- **User name:** Enter the user name for the selected wireless network.
- **Password:** This is the password for the network.

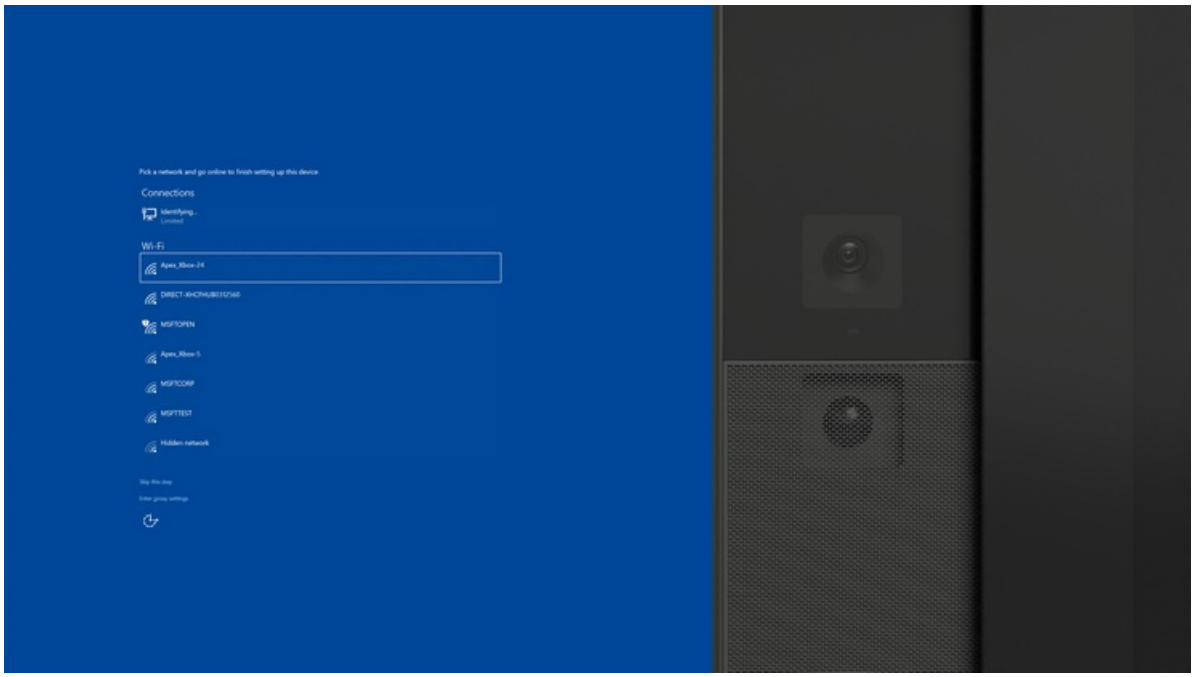
What happens?

The device will attempt to connect to the specified network. If it's successful, you'll be taken to the Set up for you page.

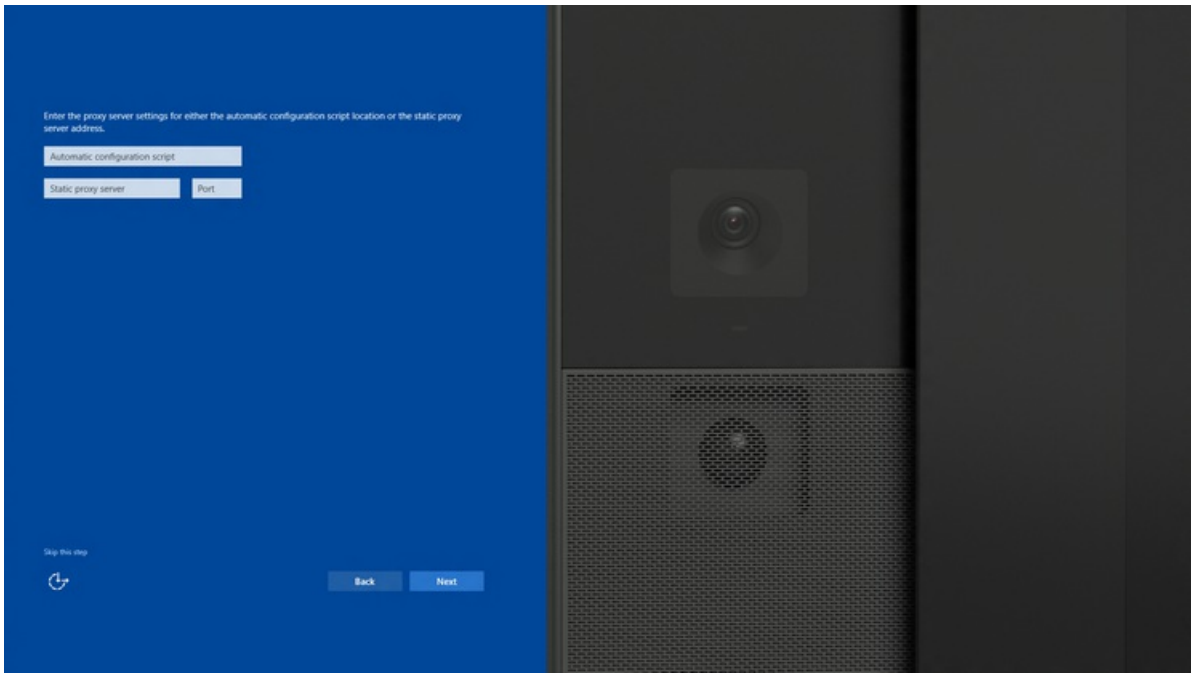
Network proxy setup

This page will be shown when the device detects a wired connection with limited connectivity. You have three options:

- You can select a wireless network to use instead of the limited wired connection.
- You can skip connecting to a network by selecting **Skip this step**. You'll be taken to the Set up for you page. **Note** If you skip this, the device will not have a network connection, and nothing that requires a network connection will work on your Surface Hub, including things like email and calendar synchronization. You can connect to a wireless network later using Settings (see Wireless network managment).
- You can select **Enter proxy settings** which will allow you to specify how to use the network proxy. You'll be taken to the next screen.



This is the screen you'll see if you clicked **Enter proxy settings** on the previous screen.



Details

In order to make a network connection, you'll need to fill in either a script name, or the proxy server and port info.

- **Proxy script:** Provide the address of a proxy script.
- **Proxy server and port:** You can provide the proxy server address and port.

What happens?

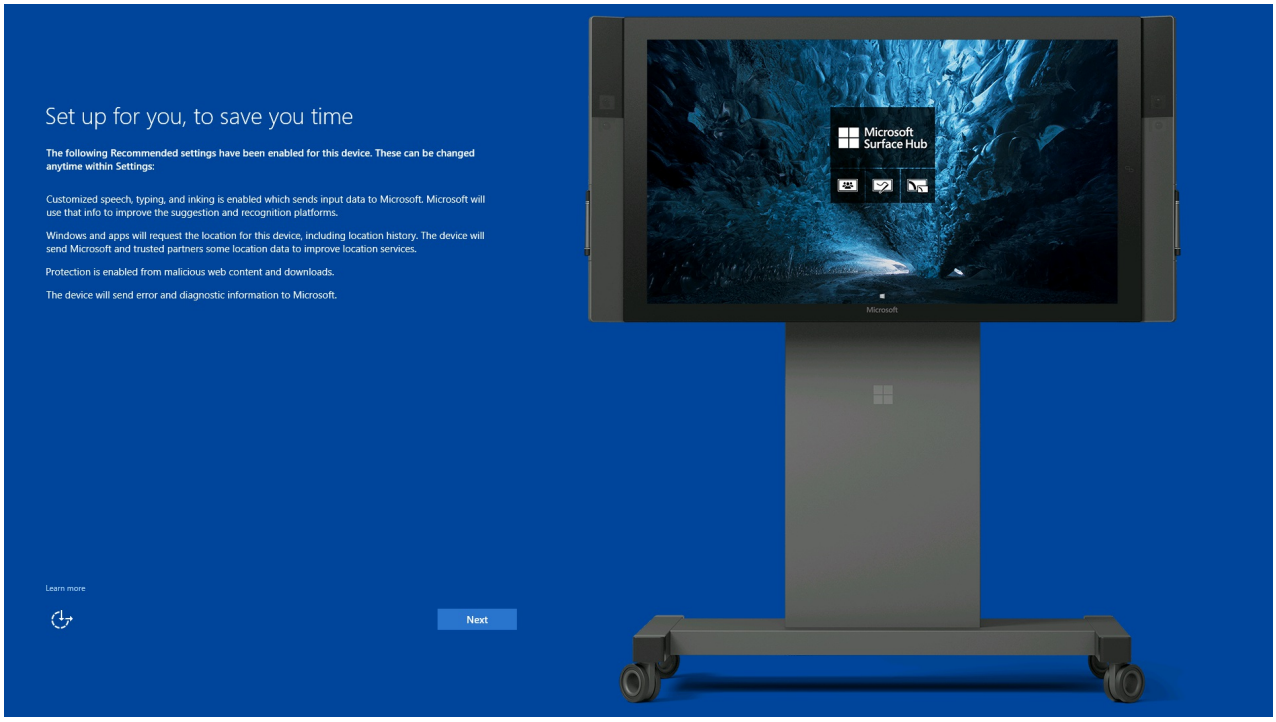
When you click **Next**, the device will attempt to connect to the proxy server. If successful, you'll be taken to the Set up for you page.

You can skip connecting to a network by selecting **Skip this step**. You'll be taken to the Set up for you page.

> **Note** If you skip this, the device will not have a network connection, and nothing that requires a network connection will work on your Surface Hub, including things like email and calendar synchronization. You can connect to a wireless network later using Settings (see Wireless network managment).

Set up for you page

This screen is purely informational, and shows which recommended settings have been enabled by default.



Details

You should read this screen and note which services have been enabled by default. All of them can be changed using the Settings app if need be, but you should be careful about the effects of doing so. See Intro to Surface Hub for details.

Once you're done reviewing the settings, click **Next** to go on.

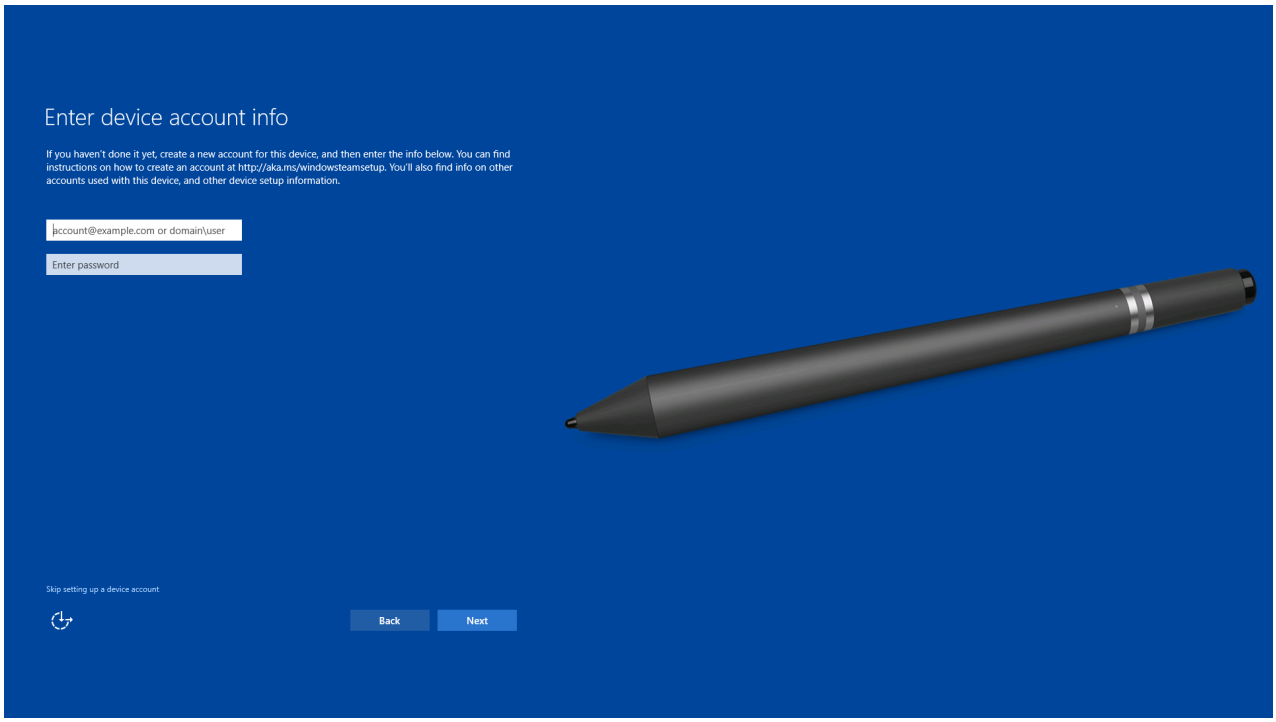
What happens?

The settings shown on the page have already been made, and can't be changed until after first run is completed.

Device account page

On this page, the Surface Hub will ask for credentials for the device account that you previously configured. (See Create and test a device account.) The Surface Hub will attempt to discover various properties of the account, and may ask for more information on another page if it does not succeed.

> **Note** This section does not cover specific errors that can happen during first run. See Troubleshoot Surface Hub for more information on errors.



Details

Use either a **user principal name (UPN)** or a **domain\user name** as the account identifier in the first entry field. Use the format that matches your environment, and enter the password.

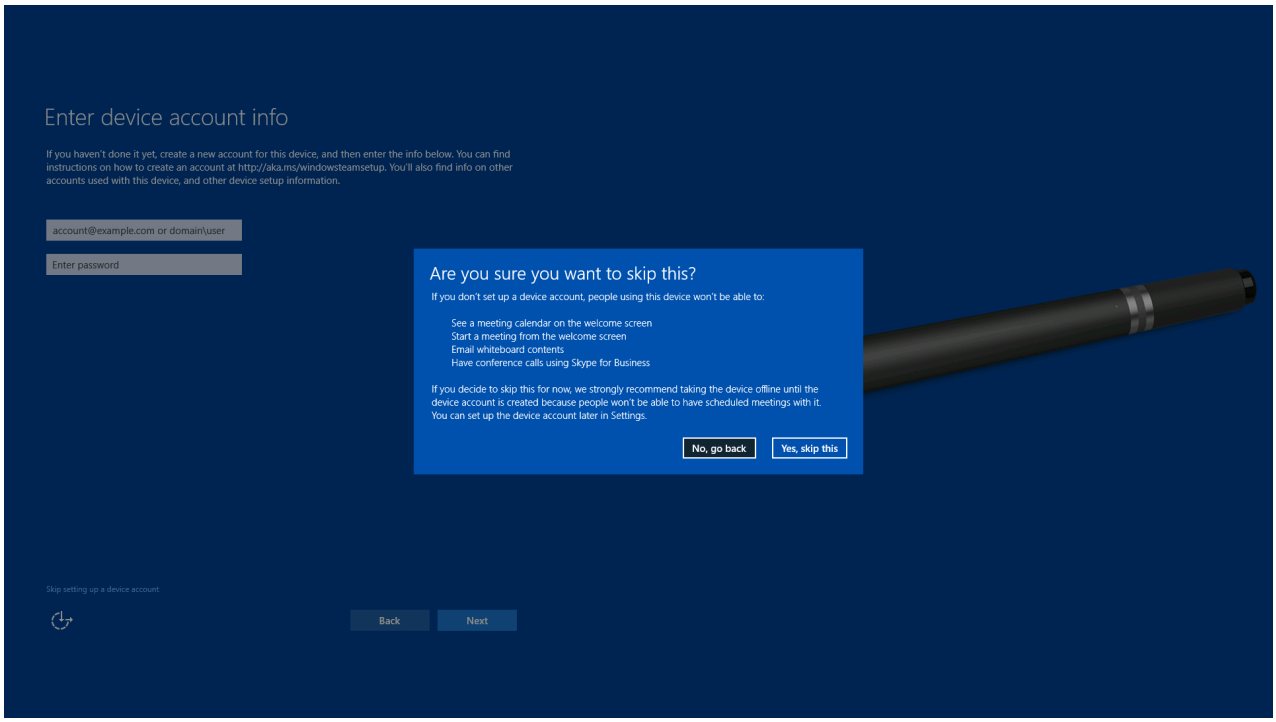
ENVIRONMENT	REQUIRED FORMAT FOR DEVICE ACCOUNT
Device account is hosted only online.	username@domain.com
Device account is hosted only on-prem.	DOMAIN\username
Device account is hosted online and on-prem (hybrid).	DOMAIN\username

Click **Skip setting up a device account** to skip setting up a device account. However, if you don't set up a device account, the device will not be fully integrated into your infrastructure. For example, people won't be able to:

- See a meeting calendar on the Welcome screen
- Start a meeting from the Welcome screen
- Email whiteboards from OneNote
- Use Skype for Business for meetings

If you skip setting it up now, you can add a device account later by using the Settings app.

If you click **Skip setting up a device account**, the device will display a dialog box showing what will happen if the device doesn't have a device account. If you choose **Yes, skip this**, you will be sent to the Name this device page.



What happens?

The device will use the UPN or DOMAIN\User name and password for the device account to do the following:

- Check if the account exists in Active Directory (AD) or Azure Active Directory (Azure AD):
  - If a UPN was entered: the device will look for the account in Azure AD.
  - If a DOMAIN\User name was entered: the device will look for the account in AD.
- Look up the Microsoft Exchange server for the account's mailbox.
- Look up the Session Initiation Protocol (SIP) address for the account.
- Pull the account's display name and alias attributes.

Exchange server page

This page will only be shown if there's a problem. Typically, it means that the device account that you provided was found in Active Directory (AD) or Azure Active Directory (Azure AD), but the Exchange server for the account was not discovered.

Details

Enter the name of the Exchange server where the device account's mailbox is hosted.

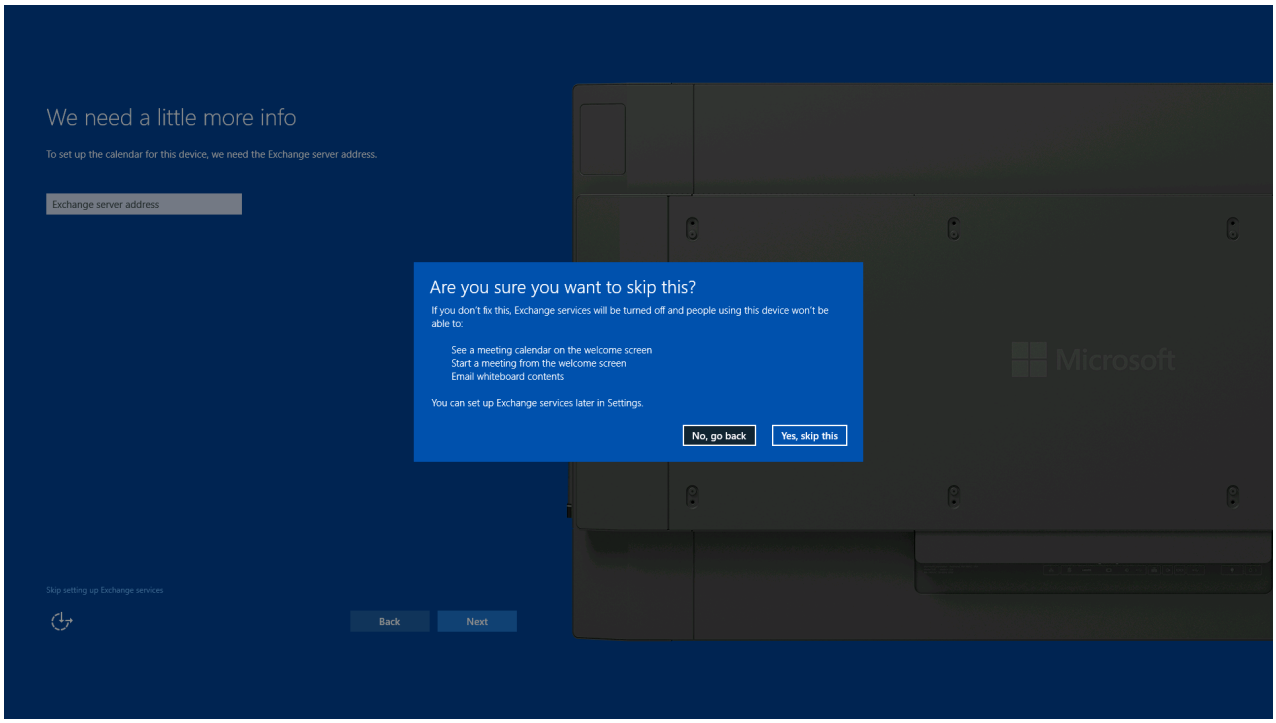
Click **Skip setting up Exchange services** to skip this step. If you do, people will not be able to:

- See a meeting calendar on the welcome screen.
- Start a meeting from the welcome screen.
- Email whiteboards from OneNote.

See Intro to Surface Hub for details on setup dependencies.

You can enable Exchange services for a device account later by using the Settings app.

If you click **Skip setting up Exchange services**, the device will display a dialog showing what will happen. If you choose **Yes, skip this**, then Exchange services will not be set up.



What happens?

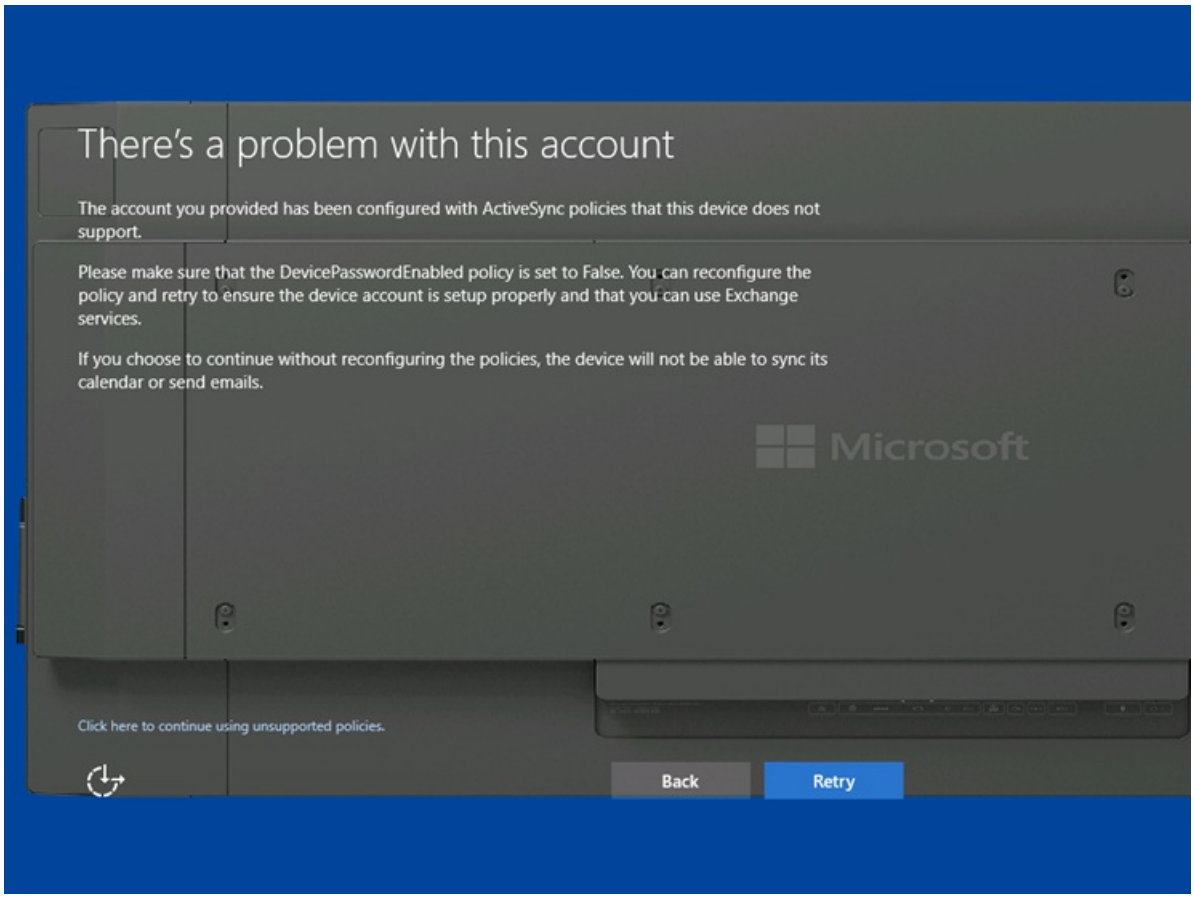
The Surface Hub will attempt to validate the device account on the Exchange server that you enter here. If the Exchange server can be reached and validates, then first run will proceed.

If you choose to skip setting up Exchange services, the Surface Hub will stop looking for the Exchange server, and no Exchange services (mail and calendar) will be enabled.

Exchange policies page

This page will be shown when:

- The device account is using an Exchange Active Sync (EAS) policy where the PasswordEnabled policy is set to 1.
- There's no connection to Exchange.
- Exchange returns a status code indicating an error. (For example: The account has been provisioned to too many devices.)
- Exchange supported protocols are not supported by the Surface Hub.
- Exchange returns incorrect XML.



Details

This page is purely informational, so no input is required. However, you have two options for proceeding: either skipping ahead or retrying the validation that caused the error. Before deciding which option is best, please read the following **What happens?** section. You may be able to fix the problem elsewhere before you click on one of the options.

- **Click here to continue using unsupported policies**: click on this to continue first run. The Surface Hub will not be able to use Exchange services, or sync.
- **Retry**: check the policy on the Exchange server again.

What happens?

The Surface Hub checks whether the device account's EAS policy has the PasswordEnabled policy set to 0 (False). If this is not the case, mail and calendar can't be synced and the Surface Hub can't use any Exchange services. You

can use your Exchange management tools from a PC to check that the device account has the PasswordEnabled policy set to 0. If that's not the case, you can reconfigure the account and click **Retry** here.

If the policy has already been configured properly, check that your device is properly connected to the network or Internet, and can reach your Exchange server, because this page will also be shown if the Surface Hub can't reach the Exchange server.

Another possible reason for not being able to reach Exchange is because of certificate-based authentication. You may wind up on this page because of certificate issues. Note that if the device displays error codes 0x80072F0D or 0X800C0019, then a certificate is required. Because provisioning is done on the first page of the first run process, you must disable Exchange services by clicking **Click here to continue using unsupported policies**, and then install the correct certificates through the Settings app.

If you choose to skip this check, the Surface Hub will stop looking for the Exchange server and validating EAS policies, and no Exchange services will be enabled. See Intro to Surface Hub for details on setup dependencies.

Name this device page

This page asks you to provide two names that will be used for identifying the Surface Hub.

Details

If the default values shown are correct, then you can click **Next** to go on. Otherwise, enter data in one or both of the text boxes.

- **Friendly name:** This is the name that people will see when they want to wirelessly connect to the Surface Hub.
- **Device name:** Can be set to any unique name as described on the screen.

As long as both names are within the length requirements and do not use restricted characters, clicking **Next** will take you to the next page, Set up admins for this device.

What happens?

The Surface Hub requires two names for the device, which will default to:

- **Friendly name:** Defaults to the Display Name of the device account
- **Device name:** Defaults to the alias of the device account

While either of the names can be changed later, keep in mind that:

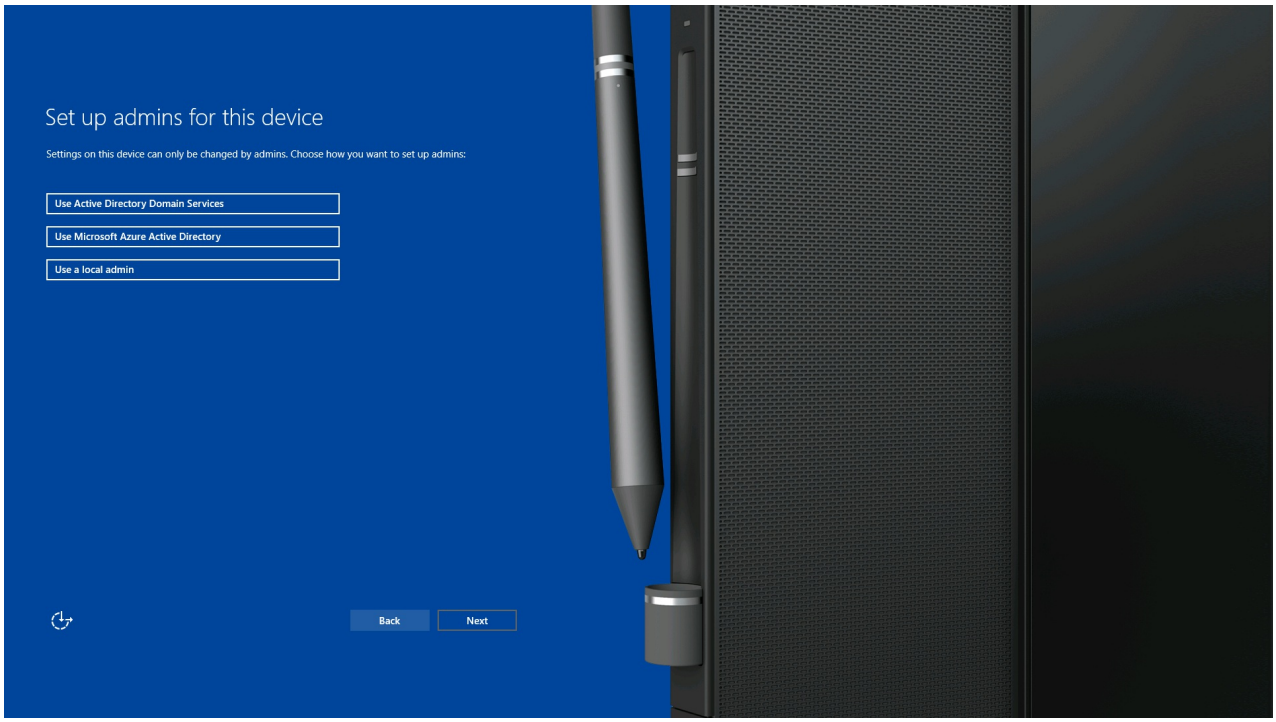
- The friendly name should be recognizable and different so that people can distinguish one Surface Hub from another when trying to wirelessly connect.
- If you decide to domain join the device, the device name must not be the same as any other device on the account's Active Directory domain. The device can't join the domain if it is using the same name as another domain-joined device.

Set up admins for this device page

On this page, you will choose from several options for how you want to set up admin accounts to locally manage your device.

Because every Surface Hub can be used by any number of authenticated employees, settings are locked down so that they can't change from session to session. Only admins can configure the settings on the device, and on this page, you'll choose which type of admins have that privilege.

> **Note** The purpose of this page is primarily to determine who can configure the device from the device's UI; that is, who can actually visit a device, log in, open up the Settings app, and make changes to the Settings.



Details

Choose one of the three available options:

- **Use Microsoft Azure Active Directory**
- **Use Active Directory Domain Services**
- **Use a local admin**

What happens?

This is what happens when you choose an option.

- **Use Microsoft Azure Active Directory**

  Clicking this option allows you to join the device to Azure AD. Once you click **Next**, the device will restart to apply some settings, and then you'll be taken to the Use Microsoft Azure Active Directory page and asked to enter credentials that can allow you to join Azure AD. After joining, admins from the joined organization will be able to use the Settings app. The specific people that will be allowed depends on your Azure AD subscription and how you've configured the settings for your Azure AD organization.

- **Use Active Directory Domain Services**

  Click this option to join the device to AD. Once you click **Next**, you'll be taken to the Use Active Directory Domain Services page and asked to enter credentials that allow you to join the specified domain. After joining, you can pick a security group from the joined domain, and people from that security group will be able to use the Settings app.

- **Use a local admin**

  Choosing this option will allow you to create a single local admin. This admin won't be backed by any directory service, so we recommend you only choose this case if the device does not have access to Azure AD or AD. Once you create an admin's user name and password on the Use a local admin page, you will need to re-enter those same credentials whenever you open the Settings app.

  Note that a local admin must have physical access to the Surface Hub to log in.

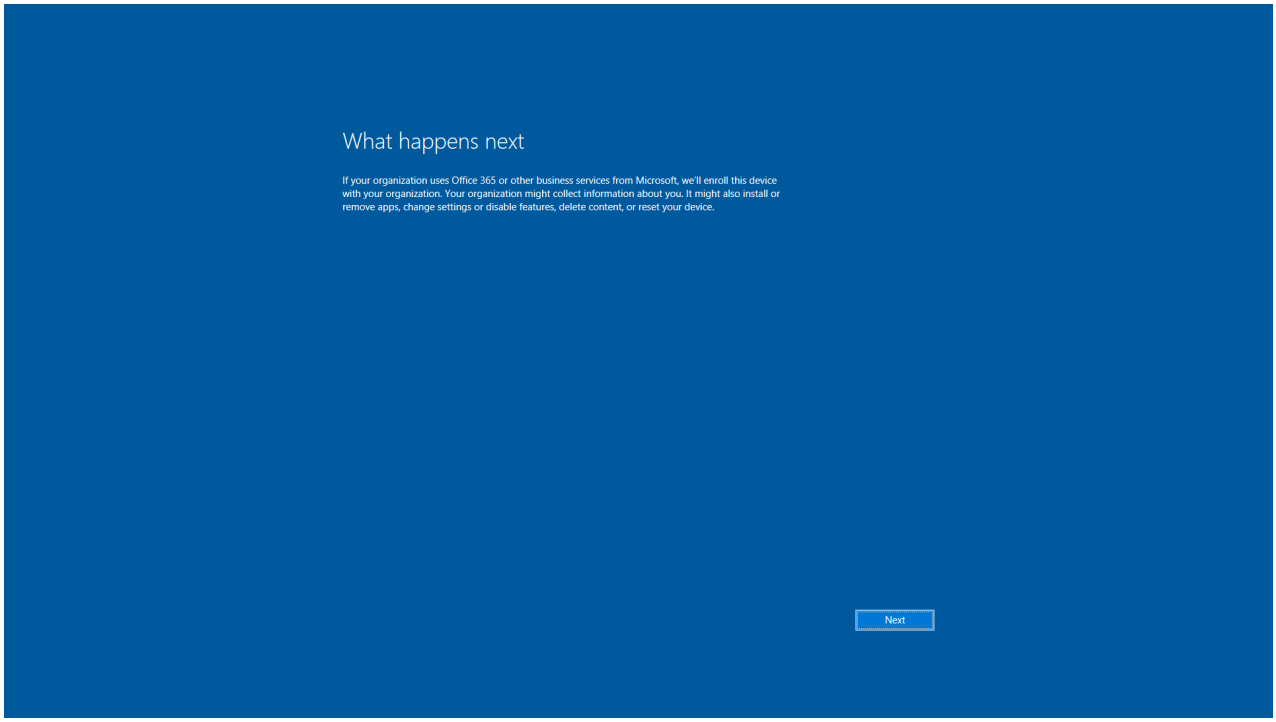
> **Note** After you finish this process, you won't be able to change the device's admin option unless you reset the device.

Use Microsoft Azure Active Directory

If you've decided to join your Surface Hub to Azure Active Directory (Azure AD), you'll see this **What happens next** page. Read it and click **Next** to go to the **Let's get you signed in page**.

Joining Azure AD has two primary benefits:

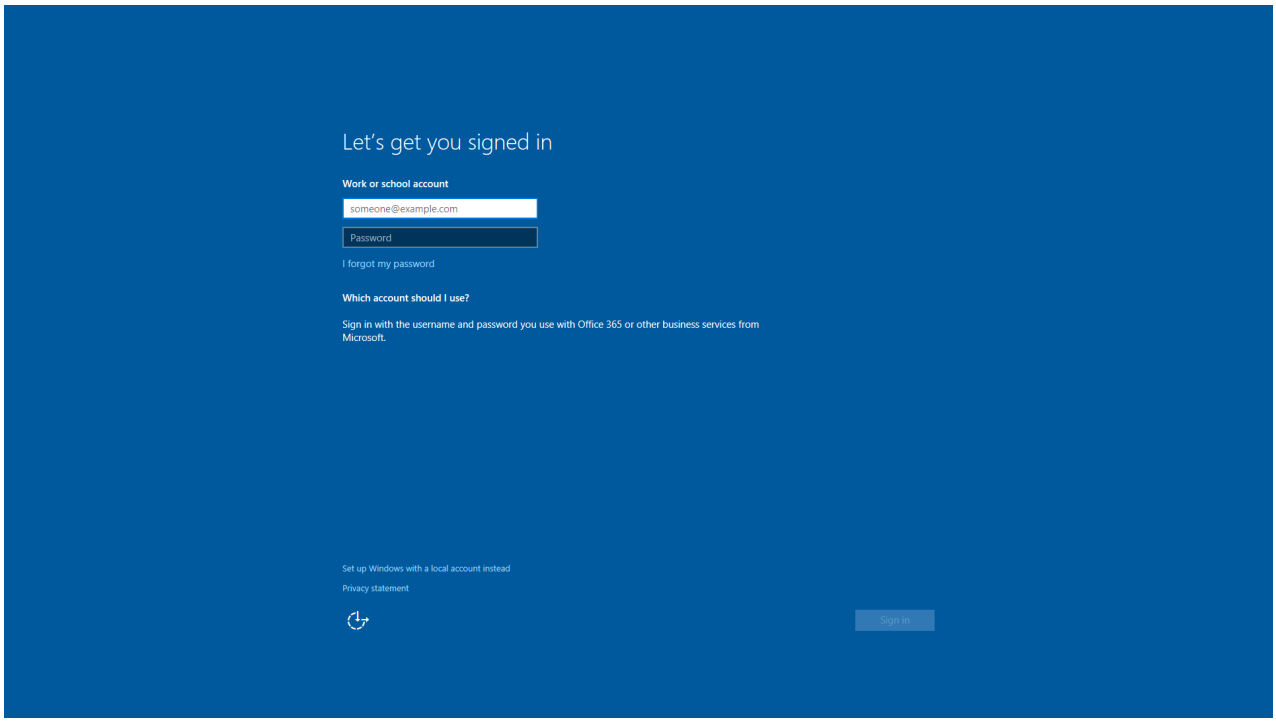
1. Some employees from your organization will be able to access the device as admins, and will be able to start the Settings app and configure the device. People that have admin permissions will be defined in your Azure AD subscription.
2. If your Azure AD is connected to a mobile device management (MDM) solution, the device will enroll with that MDM solution so you can apply policies and configuration.



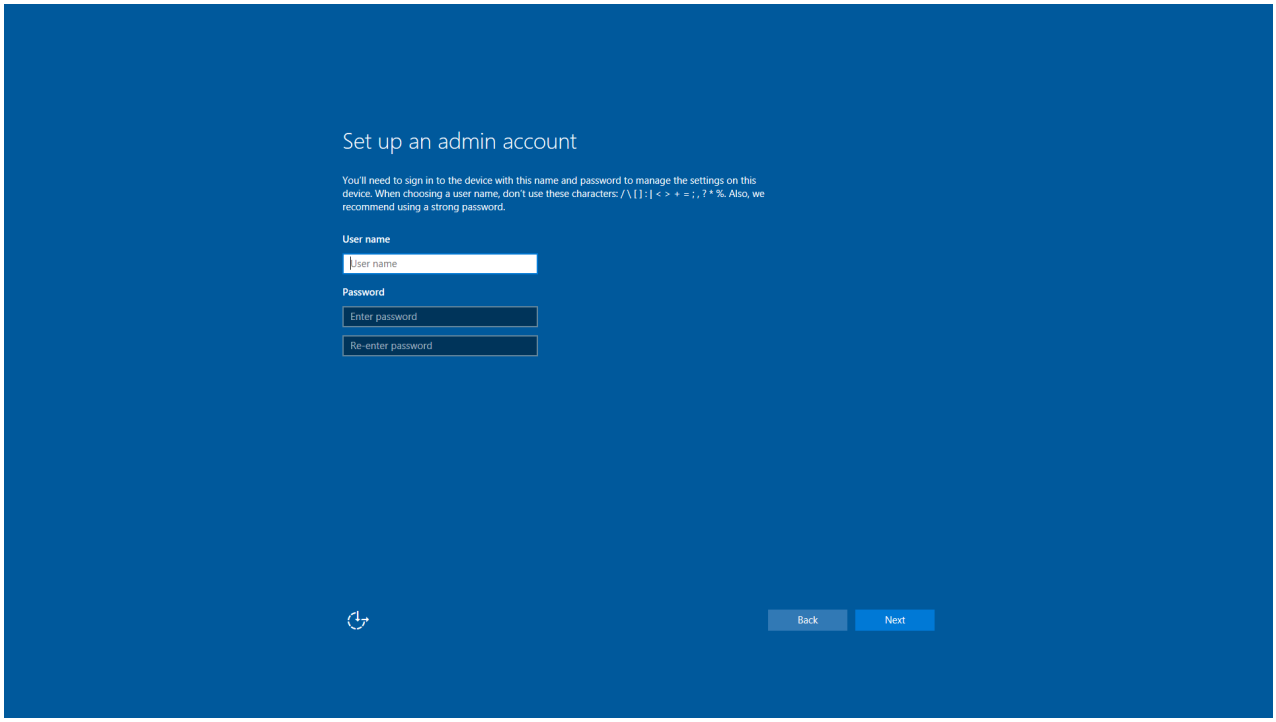
Details

The following input is required:

- **User's UPN:** The user principal name (UPN) of an account that can join Azure AD.
- **Password:** The password of the account you're using to join Azure AD.



If you get to this point and don't have valid credentials for an Azure AD account, the device will allow you to continue by creating a local admin account. Click **Set up Windows with a local account instead**.



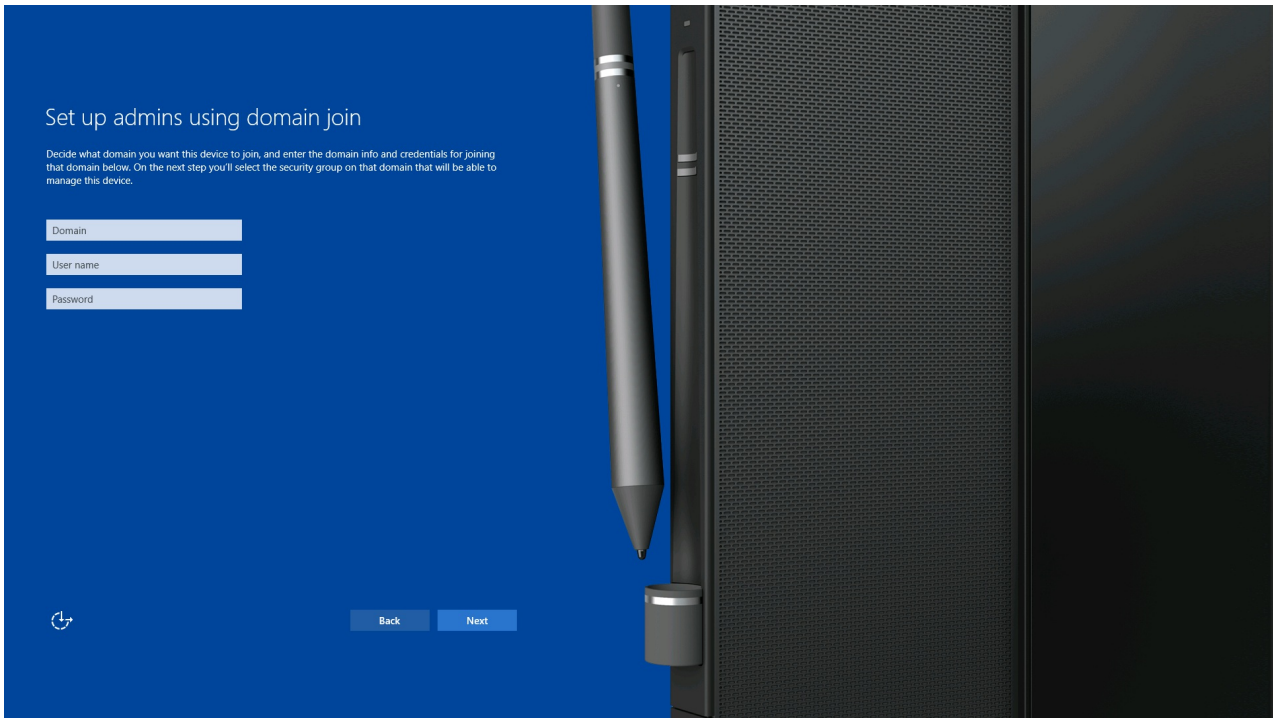
What happens?

Once you enter valid Azure AD account credentials, the device will try to join the associated Azure AD organization. If this succeeds, then the device will provision employees in that organization to be local admins on the device. If your Azure AD tenant was configured for it, the device will also enroll into MDM.

Use Active Directory Domain Services

This page will ask for credentials to join a domain so that the Surface Hub can provision a security group as administrators of the device.

Once the device has been domain joined, you must specify a security group from the domain you joined. This security group will be provisioned as administrators on the Surface Hub, and anyone from the security group can enter their domain credentials to access Settings.

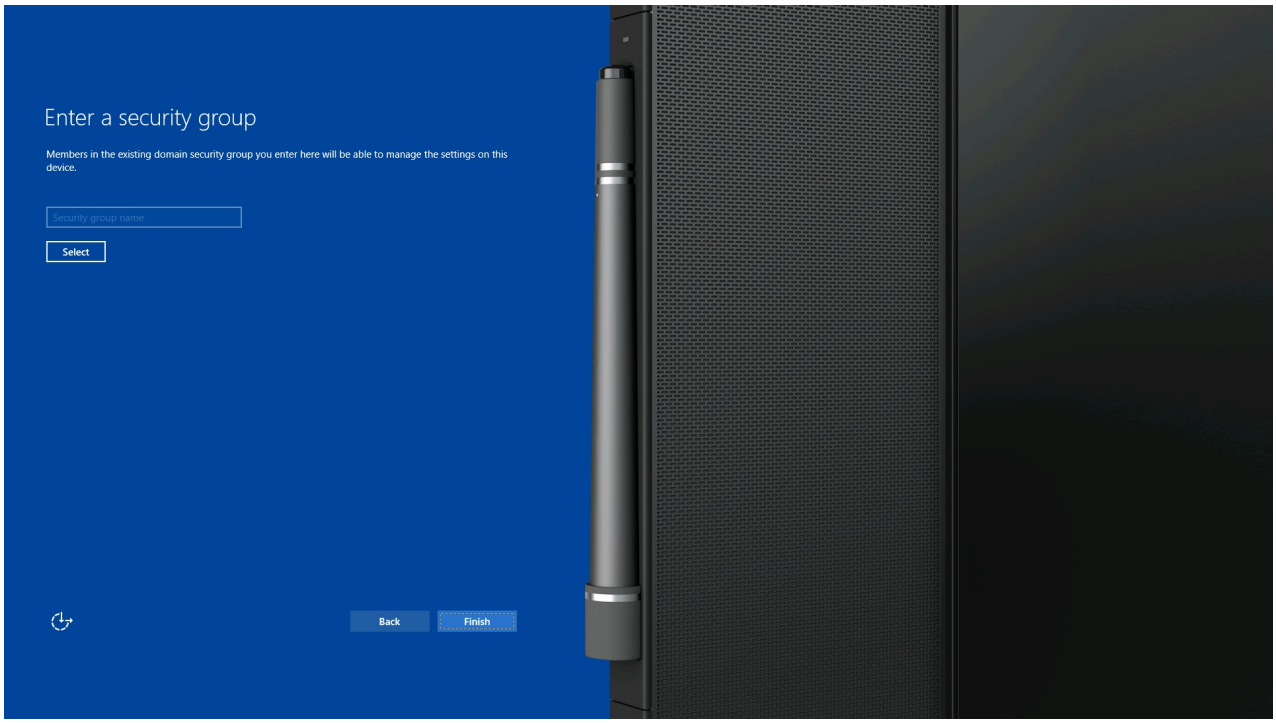


Details

The following input is required:

- **Domain:** This is the fully qualified domain name (FQDN) of the domain that you want to join. A security group from this domain can be used to manage the device.
- **User name:** The user name of an account that has sufficient permission to join the specified domain.
- **Password:** The password for the account.

After the credentials are verified, you will be asked to type a security group name. This input is required.



What happens?

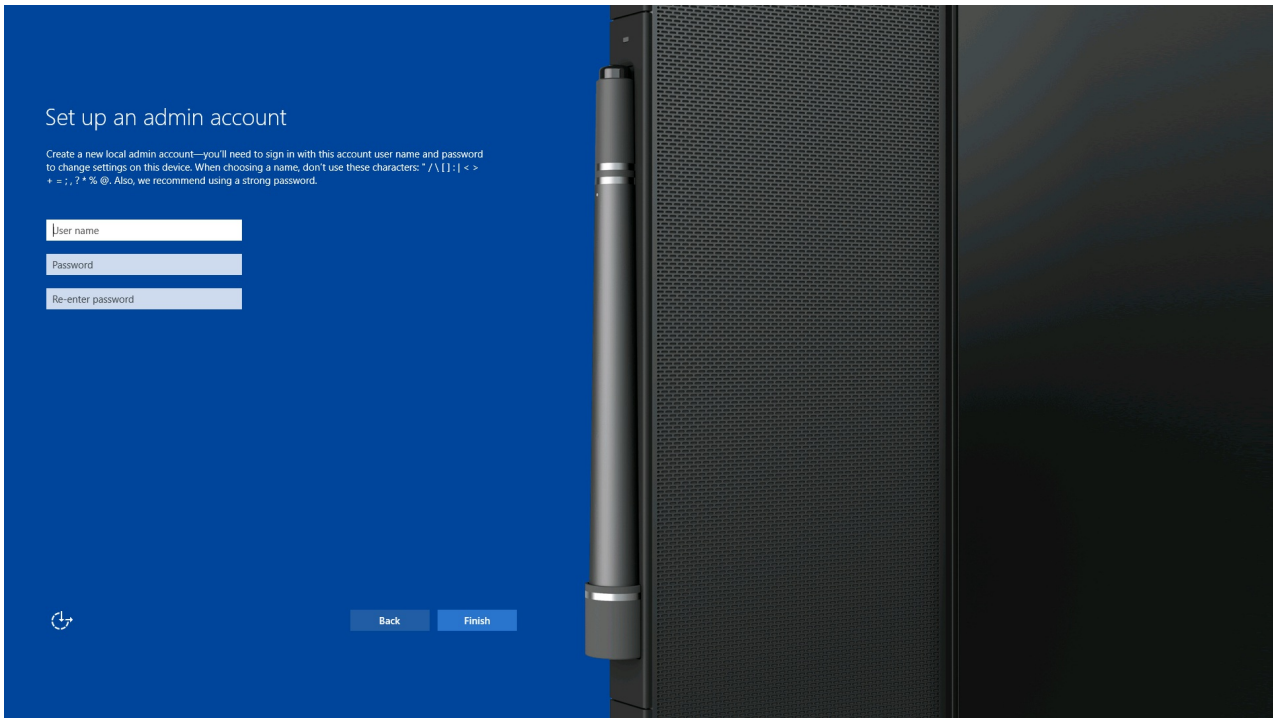
Using the provided domain, account credentials from the Use Active Directory Domain Services page and the device name from the Name this device page, the Surface Hub will attempt to join the domain. If the join is successful, first run will continue, and will ask for a security group. If the join is not successful, first run will halt and ask you to change the information provided.

If the join is successful, you'll see the **Enter a security group** page. When you click the **Select** button on this page, the device will search for the specified security group on your domain. If found, the group will be verified. Click **Finish** to complete the first run process.

> **Note** If you domain join the Surface Hub, you can't unjoin the device without resetting it.

Use a local admin

If you decide not to use Azure Active Directory (Azure AD) or Active Directory (AD) to manage the Surface Hub, you'll need to create a local admin account.



Details

The following input is required:

- **User name:** This is the user name of the local admin account that will be created for this Surface Hub.
- **Password:** This is the password of the device account.
- **Re-enter password:** Verifying the password as in the previous box.

What happens?

This page will attempt to create a new admin account using the credentials that you enter here. If it's successful, then first run will end. If not, you'll be asked for different credentials.

Update the Surface Hub

> **Important** Before you do the updates, make sure you read Save your BitLocker key in order to make sure you have a backup of the key.

In order to get the latest features and fixes, you should update your Surface Hub as soon as you finish all of the preceding first-run steps.

1. Make sure the device has access to the Windows Update servers or to Windows Server Update Services (WSUS). To configure WSUS, see Using WSUS.
2. Open Settings, click **Update & security**, then **Windows Update**, and then click **Check for updates**.
3. If updates are available, they will be downloaded. Once downloading is complete, click the **Update now** button to install the updates.

4. Follow the onscreen prompts after the updates are installed. You may need to restart the device.

Manage Microsoft Surface Hub

5/4/2017 • 1 min to read • Edit Online

After initial setup of Microsoft Surface Hub, the device's settings and configuration can be modified or changed in a couple ways:

- **Local management** - Every Surface Hub can be configured locally using the **Settings** app on the device. To prevent unauthorized users from changing settings, the Settings app requires admin credentials to open the app. For more information, see Local management for Surface Hub settings.
- **Remote management** - Surface Hub allow IT admins to manage settings and policies using a mobile device management (MDM) provider, such as Microsoft Intune, System Center Configuration Manager, and other third-party providers. Additionally, admins can monitor Surface Hubs using Microsoft Operations Management Suite (OMS). For more information, see Manage settings with an MDM provider, and Monitor your Microsoft Surface Hub.

> **NOTE**
> These management methods are not mutually exclusive. Devices can be both locally and remotely managed if you choose. However, MDM policies and settings will overwrite any local changes when the Surface Hub syncs with the management server.

In this section

Learn about managing and updating Surface Hub.

TOPIC	DESCRIPTION
Remote Surface Hub management	Topics related to managing your Surface Hub remotely. Include install apps, managing settings with MDM and monitoring with Operations Management Suite.
Manage Surface Hub settings	Topics related to managing Surface Hub settings: accessibility, device account, device reset, fully qualified domain name, Windows Update settings, and wireless network
Install apps on your Surface Hub	Admins can install apps can from either the Microsoft Store or the Microsoft Store for Business.
End a meeting with I'm done	At the end of a meeting, users can tap I'm Done to clean up any sensitive data and prepare the device for the next meeting.
Save your BitLocker key	Every Surface Hub is automatically set up with BitLocker drive encryption software. Microsoft strongly recommends that you make sure you back up your BitLocker recovery keys.
Connect other devices and display with Surface Hub	You can connect other device to your Surface Hub to display content.
Using a room control system	Room control systems can be used with your Microsoft Surface Hub.

Remote Surface Hub management

5/4/2017 • 1 min to read • Edit Online

In this section

TOPIC	DESCRIPTION
Manage settings with an MDM provider	Surface Hub provides an enterprise management solution to help IT administrators manage policies and business applications on these devices using a mobile device management (MDM) solution.
Monitor your Surface Hub	Monitoring for Surface Hub devices is enabled through Microsoft Operations Management Suite.
Windows updates	You can manage Windows updates on your Surface Hub by setting the maintenance window, deferring updates, or using WSUS.

Manage settings with an MDM provider (Surface Hub)

5/4/2017 • 12 min to read • Edit Online

Surface Hub and other Windows 10 devices allow IT administrators to manage settings and policies using a mobile device management (MDM) provider. A built-in management component communicates with the management server, so there is no need to install additional clients on the device. For more information, see Windows 10 mobile device management.

Surface Hub has been validated with Microsoft's first-party MDM providers:

- On-premises MDM with System Center Configuration Manager (beginning in version 1602)
- Hybrid MDM with System Center Configuration Manager and Microsoft Intune
- Microsoft Intune standalone

You can also manage Surface Hubs using any third-party MDM provider that can communicate with Windows 10 using the MDM protocol.

Enroll a Surface Hub into MDM

You can enroll your Surface Hubs using bulk or manual enrollment.

> **NOTE**
> You can join your Surface Hub to Azure Active Directory (Azure AD) to manage admin groups on the device. However, Surface Hub does not currently support automatic enrollment to Microsoft Intune through Azure AD join. If your organization automatically enrolls Azure AD-joined devices into Intune, you must disable this policy for Surface Hub before joining the device to Azure AD.
>
> **To enable automatic enrollment for Microsoft Intune**
>
> 1. In the Azure classic portal, navigate to the **Active Directory** node and select your directory.
> 2. Click the **Applications** tab, then click **Microsoft Intune**.
> 3. Under **Manage devices for these users**, click **Groups**.
> 4. Click **Select Groups**, then select the groups of users you want to automatically enroll into Intune. **Do not include accounts that are used to enroll Surface Hubs into Intune.**
> 5. Click the checkmark button, then click **Save**.

Bulk enrollment

To configure bulk enrollment

- Surface Hub supports the Provisioning CSP for bulk enrollment into MDM. For more information, see Windows 10 bulk enrollment.
  --OR--
- If you have an on-premises System Center Configuration Manager infrastructure, see How to bulk enroll devices with On-premises Mobile Device Management in System Center Configuration Manager.

Manual enrollment

To configure manual enrollment

1. On your Surface Hub, open **Settings**.

2. Type the device admin credentials when prompted.
3. Select **This device**, and navigate to **Device management**.
4. Under **Device management**, select **+ Device management**.
5. Follow the instructions in the dialog to connect to your MDM provider.

Manage Surface Hub settings with MDM

You can use MDM to manage some Surface Hub CSP settings, and some Windows 10 settings. Depending on the MDM provider that you use, you may set these settings using a built-in user interface, or by deploying custom SyncML. Microsoft Intune and System Center Configuration Manager provide built-in experiences to help create policy templates for Surface Hub. Refer to documentation from your MDM provider to learn how to create and deploy SyncML.

Supported Surface Hub CSP settings

You can configure the Surface Hub settings in the following table using MDM. The table identifies if the setting is supported with Microsoft Intune, System Center Configuration Manager, or SyncML.

For more information, see SurfaceHub configuration service provider.

SETTING	NODE IN THE SURFACEHUB CSP	SUPPORTED WITH INTUNE?	SUPPORTED WITH CONFIGURATION MANAGER?	SUPPORTED WITH SYNCML*?
Maintenance hours	MaintenanceHoursSimple/Hours/StartTime MaintenanceHoursSimple/Hours/Duration	Yes	Yes	Yes
Automatically turn on the screen using motion sensors	InBoxApps/Welcome/AutoWakeScreen	Yes	Yes	Yes
Require a pin for wireless projection	InBoxApps/WirelessProjection/PINRequired	Yes	Yes	Yes
Enable wireless projection	InBoxApps/WirelessProjection/Enabled	Yes	Yes. Use a custom setting.	Yes
Miracast channel to use for wireless projection	InBoxApps/WirelessProjection/Channel	Yes	Yes. Use a custom setting.	Yes
Connect to your Operations Management Suite workspace	MOMAgent/WorkspaceID MOMAgent/WorkspaceKey	Yes	Yes. Use a custom setting.	Yes
Welcome screen background image	InBoxApps/Welcome/CurrentBackgroundPath	Yes	Yes. Use a custom setting.	Yes
Meeting information displayed on the welcome screen	InBoxApps/Welcome/MeetingInfoOption	Yes	Yes. Use a custom setting.	Yes
Friendly name for wireless projection	Properties/FriendlyName	Yes. Use a custom policy.)	Yes. Use a custom setting.	Yes

SETTING	NODE IN THE SURFACEHUB CSP	SUPPORTED WITH INTUNE?	SUPPORTED WITH CONFIGURATION MANAGER?	SUPPORTED WITH SYNCML*?
Device account, including password rotation	DeviceAccount/ `<name_of_policy>` See SurfaceHub CSP.	No	No	Yes

*Settings supported with SyncML can also be configured in a Windows Imaging and Configuration Designer (Windows ICD) provisioning package.

Supported Windows 10 settings

In addition to Surface Hub-specific settings, there are numerous settings common to all Windows 10 devices. These settings are defined in the Configuration service provider reference.

The following tables include info on Windows 10 settings that have been validated with Surface Hub. There is a table with settings for these areas: security, browser, Windows Updates, Windows Defender, remote reboot, certificates, and logs. Each table identifies if the setting is supported with Microsoft Intune, System Center Configuration Manager, or SyncML.

Security settings

SETTING	DETAILS	CSP REFERENCE	SUPPORTED WITH INTUNE?	SUPPORTED WITH CONFIGURATION MANAGER?	SUPPORTED WITH SYNCML*?
Allow Bluetooth	Keep this enabled to support Bluetooth peripherals.	Connectivity/AllowBluetooth	Yes. Use a custom policy.	Yes. Use a custom setting.	Yes
Bluetooth policies	Use to set the Bluetooth device name, and block advertising, discovery, and automatic pairing.	Bluetooth/ `<name of policy>` See Policy CSP	Yes. Use a custom policy.	Yes. Use a custom setting.	Yes
Allow camera	Keep this enabled for Skype for Business.	Camera/AllowCamera	Yes. Use a custom policy.	Yes. Use a custom setting.	Yes
Allow location	Keep this enabled to support apps such as Maps.	System/AllowLocation	Yes. Use a custom policy.	Yes. Use a custom setting.	Yes
Allow telemetry	Keep this enabled to help Microsoft improve Surface Hub.	System/AllowTelemetry	Yes. Use a custom policy.	Yes. Use a custom setting.	Yes

*Settings supported with SyncML can also be configured in a Windows Imaging and Configuration Designer (Windows ICD) provisioning package.

Browser settings

SETTING	DETAILS	CSP REFERENCE	SUPPORTED WITH INTUNE?	SUPPORTED WITH CONFIGURATION MANAGER?	SUPPORTED WITH SYNCML*?
Homepages	Use to configure the default homepages in Microsoft Edge.	Browser/Homepages	Yes. Use a custom policy.	Yes. Use a custom setting.	Yes
Allow cookies	Surface Hub automatically deletes cookies at the end of a session. Use this to block cookies within a session.	Browser/AllowCookies	Yes. Use a custom policy.	Yes. Use a custom setting.	Yes
Allow developer tools	Use to stop users from using F12 Developer Tools.	Browser/AllowDeveloperTools	Yes. Use a custom policy.	Yes. Use a custom setting.	Yes
Allow Do Not Track	Use to enable Do Not Track headers.	Browser/AllowDoNotTrack	Yes. Use a custom policy.	Yes. Use a custom setting.	Yes
Allow pop-ups	Use to block pop-up browser windows.	Browser/AllowPopups	Yes. Use a custom policy.	Yes. Use a custom setting.	Yes
Allow search suggestions	Use to block search suggestions in the address bar.	Browser/AllowSearchSuggestionsinAddressBar	Yes. Use a custom policy.	Yes. Use a custom setting.	Yes
Allow SmartScreen	Keep this enabled to turn on SmartScreen.	Browser/AllowSmartScreen	Yes. Use a custom policy.	Yes. Use a custom setting.	Yes
Prevent ignoring SmartScreen Filter warnings for websites	For extra security, use to stop users from ignoring SmartScreen Filter warnings and block them from accessing potentially malicious websites.	Browser/PreventSmartScreenPromptOverride	Yes. Use a custom policy.	Yes. Use a custom setting.	Yes
Prevent ignoring SmartScreen Filter warnings for files	For extra security, use to stop users from ignoring SmartScreen Filter warnings and block them from downloading unverified files from Microsoft Edge.	Browser/PreventSmartScreenPromptOverrideForFiles	Yes. Use a custom policy.	Yes. Use a custom setting.	Yes

*Settings supported with SyncML can also be configured in a Windows Imaging and Configuration Designer (Windows ICD) provisioning package.

Windows Update settings

SETTING	DETAILS	CSP REFERENCE	SUPPORTED WITH INTUNE?	SUPPORTED WITH CONFIGURATION MANAGER?	SUPPORTED WITH SYNCML*?
Use Current Branch or Current Branch for Business	Use to configure Windows Update for Business – see Windows updates.	Update/BranchReadinessLevel	Yes. Use a custom policy.	Yes. Use a custom setting.	Yes
Defer feature updates	See above.	Update/DeferFeatureUpdatesPeriodInDays	Yes. Use a custom policy.	Yes. Use a custom setting.	Yes
Defer quality updates	See above.	Update/DeferQualityUpdatesPeriodInDays	Yes. Use a custom policy.	Yes. Use a custom setting.	Yes
Pause feature updates	See above.	Update/PauseFeatureUpdates	Yes. Use a custom policy.	Yes. Use a custom setting.	Yes
Pause quality updates	See above.	Update/PauseQualityUpdates	Yes. Use a custom policy.	Yes. Use a custom setting.	Yes
Configure device to use WSUS	Use to connect your Surface Hub to WSUS instead of Windows Update – see Windows updates.	Update/UpdateServiceUrl	Yes. Use a custom policy.	Yes. Use a custom setting.	Yes
Delivery optimization	Use peer-to-peer content sharing to reduce bandwidth issues during updates. See Configure Delivery Optimization for Windows 10 for details.	DeliveryOptimization/ `<name of policy>` See Policy CSP	Yes. Use a custom policy.	Yes. Use a custom setting.	Yes

*Settings supported with SyncML can also be configured in a Windows Imaging and Configuration Designer (Windows ICD) provisioning package.

Windows Defender settings

SETTING	DETAILS	CSP REFERENCE	SUPPORTED WITH INTUNE?	SUPPORTED WITH CONFIGURATION MANAGER?	SUPPORTED WITH SYNCML*?

SETTING	DETAILS	CSP REFERENCE	SUPPORTED WITH INTUNE?	SUPPORTED WITH CONFIGURATION MANAGER?	SUPPORTED WITH SYNCML*?
Defender policies	Use to configure various Defender settings, including a scheduled scan time.	Defender/ `<name of policy>` See Policy CSP	Yes. Use a custom policy.	Yes. Use a custom setting.	Yes
Defender status	Use to initiate a Defender scan, force a signature update, query any threats detected.	Defender CSP	No.	No.	Yes

*Settings supported with SyncML can also be configured in a Windows Imaging and Configuration Designer (Windows ICD) provisioning package.

Remote reboot

SETTING	DETAILS	CSP REFERENCE	SUPPORTED WITH INTUNE?	SUPPORTED WITH CONFIGURATION MANAGER?	SUPPORTED WITH SYNCML*?
Reboot the device immediately	Use in conjunction with OMS to minimize support costs – see Monitor your Microsoft Surface Hub.	./Vendor/MSFT/Reboot/RebootNow See Reboot CSP	No	No	Yes
Reboot the device at a scheduled date and time	See above.	./Vendor/MSFT/Reboot/Schedule/Single See Reboot CSP	Yes. Use a custom policy.	Yes. Use a custom setting.	Yes
Reboot the device daily at a scheduled date and time	See above.	./Vendor/MSFT/Reboot/Schedule/DailyRecurrent See Reboot CSP	Yes. Use a custom policy.	Yes. Use a custom setting.	Yes

*Settings supported with SyncML can also be configured in a Windows Imaging and Configuration Designer (Windows ICD) provisioning package.

Install certificates

SETTING	DETAILS	CSP REFERENCE	SUPPORTED WITH INTUNE?	SUPPORTED WITH CONFIGURATION MANAGER?	SUPPORTED WITH SYNCML*?
Install trusted CA certificates	Use to deploy trusted root and intermediate CA certificates.	RootCATrustedCertificates CSP	Yes. See Configure Intune certificate profiles.	Yes. See How to create certificate profiles in System Center Configuration Manager.	Yes

*Settings supported with SyncML can also be configured in a Windows Imaging and Configuration Designer (Windows ICD) provisioning package.

Collect logs

SETTING	DETAILS	CSP REFERENCE	SUPPORTED WITH INTUNE?	SUPPORTED WITH CONFIGURATION MANAGER?	SUPPORTED WITH SYNCML*?
Collect ETW logs	Use to remotely collect ETW logs from Surface Hub.	DiagnosticLog CSP	No	No	Yes

*Settings supported with SyncML can also be configured in a Windows Imaging and Configuration Designer (Windows ICD) provisioning package.

Generate OMA URIs for settings

You need to use a setting's OMA URI to create a custom policy in Intune, or a custom setting in System Center Configuration Manager.

To generate the OMA URI for any setting in the CSP documentation

1. In the CSP documentation, identify the root node of the CSP. Generally, this looks like `./Vendor/MSFT/<name of CSP>`
   For example, the root node of the SurfaceHub CSP is `./Vendor/MSFT/SurfaceHub`.
2. Identify the node path for the setting you want to use.
   For example, the node path for the setting to enable wireless projection is `InBoxApps/WirelessProjection/Enabled`.
3. Append the node path to the root node to generate the OMA URI.
   For example, the OMA URI for the setting to enable wireless projection is `./Vendor/MSFT/SurfaceHub/InBoxApps/WirelessProjection/Enabled`.

The data type is also stated in the CSP documentation. The most common data types are:

- char (String)
- int (Integer)
- bool (Boolean)

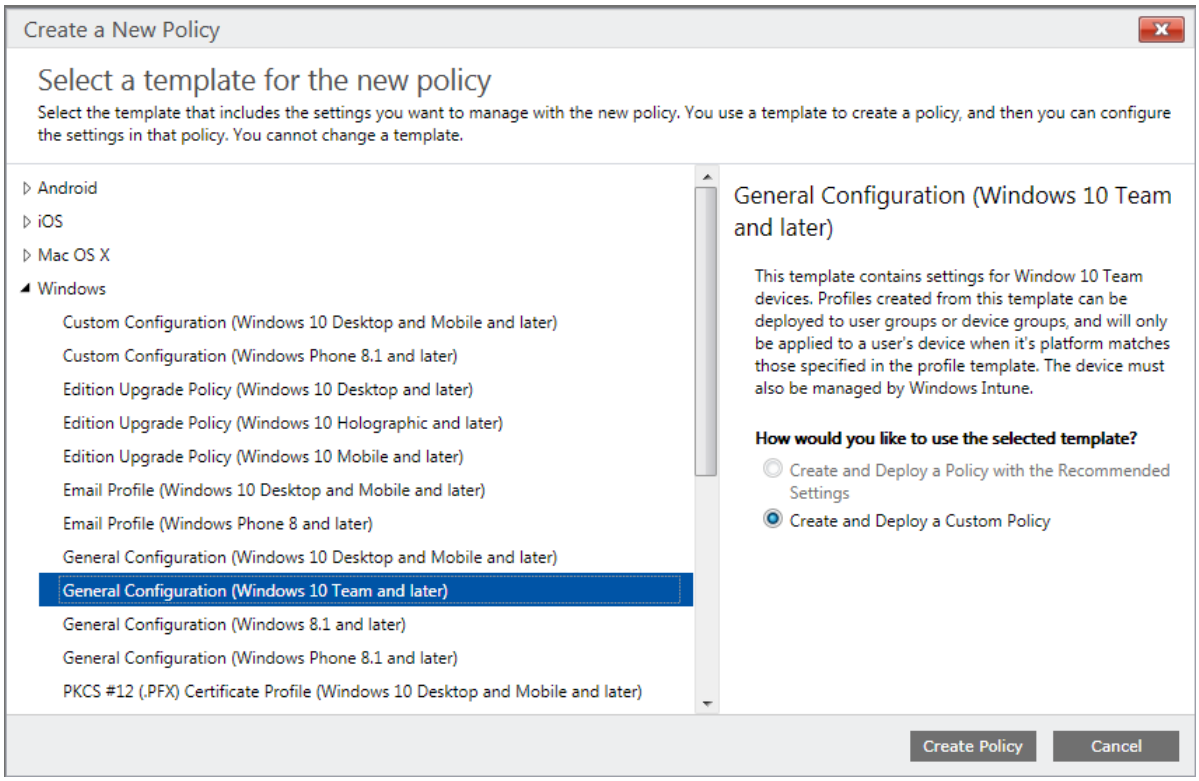
Example: Manage Surface Hub settings with Microsoft Intune

You can use Microsoft Intune to manage Surface Hub settings.

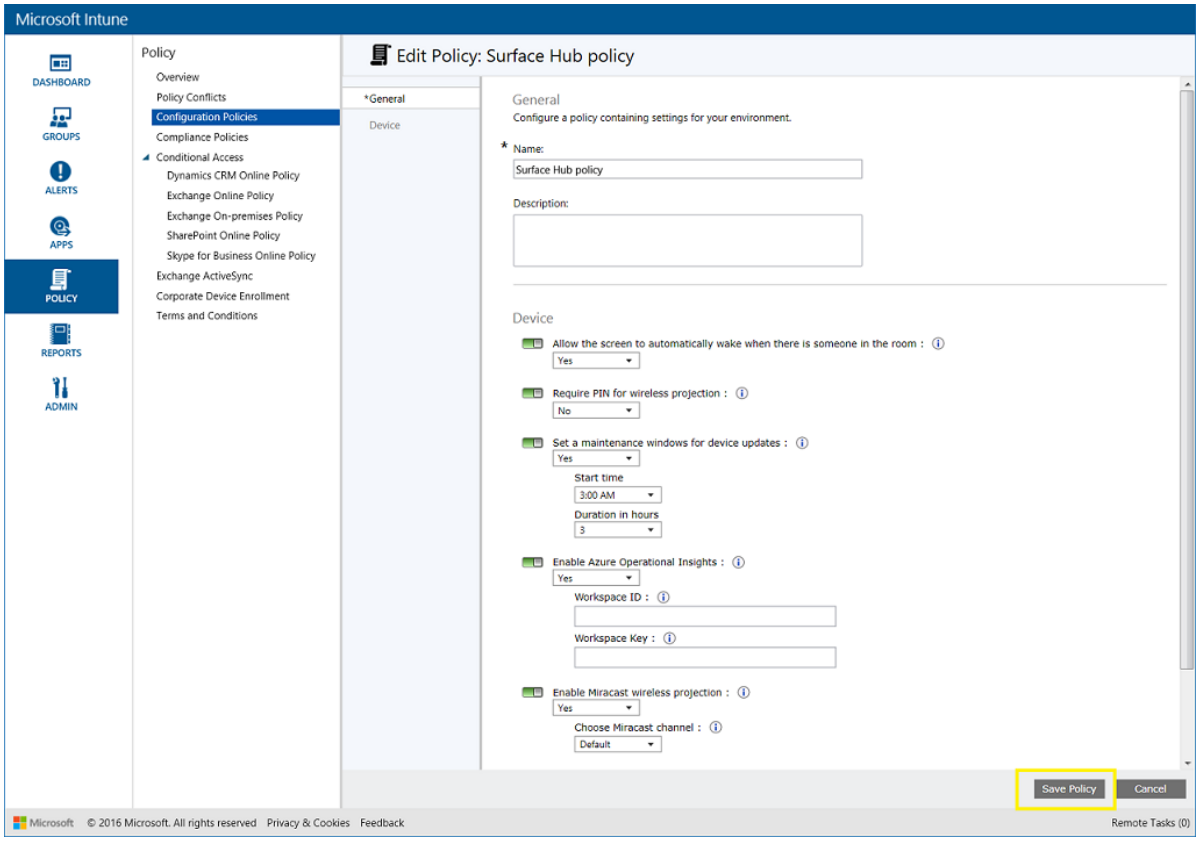
To create a configuration policy from a template

You'll use the **Windows 10 Team general configuration policy** as the template.

1. On the Intune management portal, sign in with your Intune administrator account.
2. On the left-hand navigation menu, click **Policy**.
3. In the Overview page, click **Add Policy**.
4. On **Select a template for the new policy**, expand **Windows**, select **General Configuration (Windows 10 Team and later)**, and then click **Create Policy**.



5. Configure your policy, then click **Save Policy**



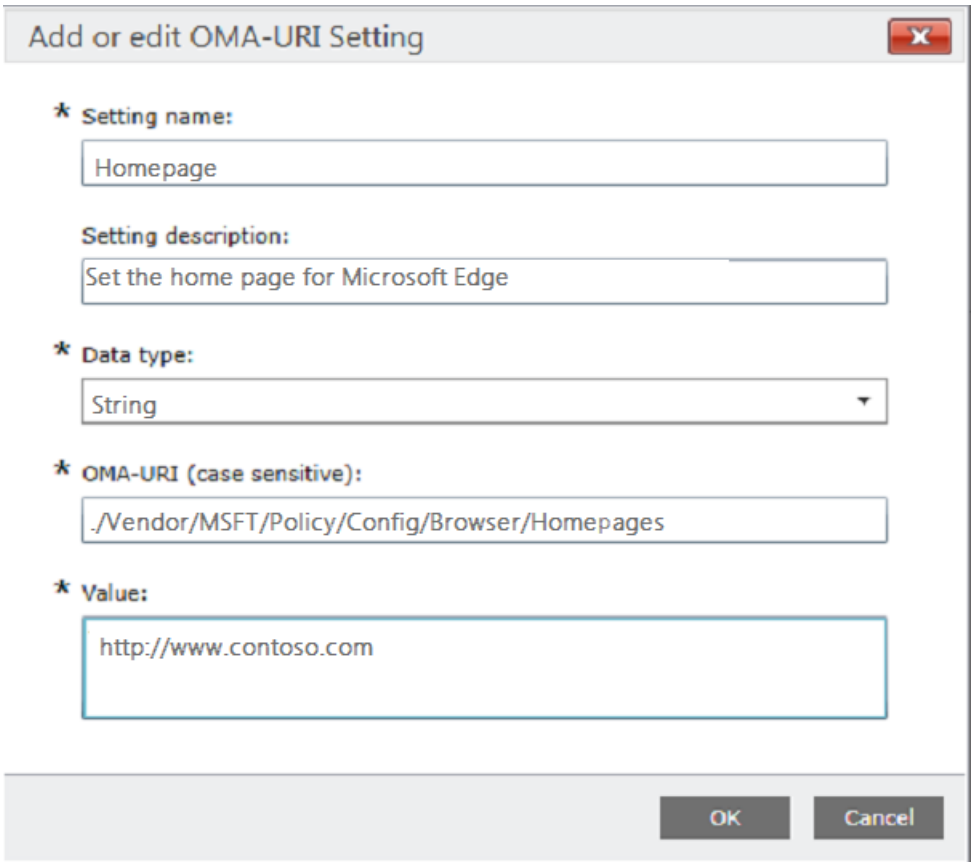
6. When prompted, click **Yes** to deploy your new policy to a user or device group. For more information, see Use groups to manage users and devices in Microsoft Intune.

To create a custom configuration policy

You'll need to create a custom policy using the **Custom Configuration (Windows 10 Desktop and Mobile and later)** template to manage settings that are not available in the **Windows 10 Team general configuration policy** template.

1. On the Intune management portal, sign in with your Intune administrator account.

2. On the left-hand navigation menu, click **Policy**.
3. On the Overview page, click **Add Policy**.
4. On **Select a template for the new policy**, expand **Windows**, select **Custom Configuration (Windows 10 Desktop and Mobile and later)**, and then click **Create Policy**.
5. Type a name and optional description for the policy.
6. Under OMA-URI Settings, click **Add**.
7. Complete the form to create a new setting, and then click **OK**.



8. Repeat Steps 6 and 7 for each setting you want to configure with this policy.
9. After you're done, click **Save Policy** and deploy it to a user or device group.

Example: Manage Surface Hub settings with System Center Configuration Manager

System Center Configuration Manager supports managing modern devices that do not require the Configuration Manager client to manage them, including Surface Hub. If you already use System Center Configuration Manager to manage other devices in your organization, you can continue to use the Configuration Manager console as your single location for managing Surface Hubs.

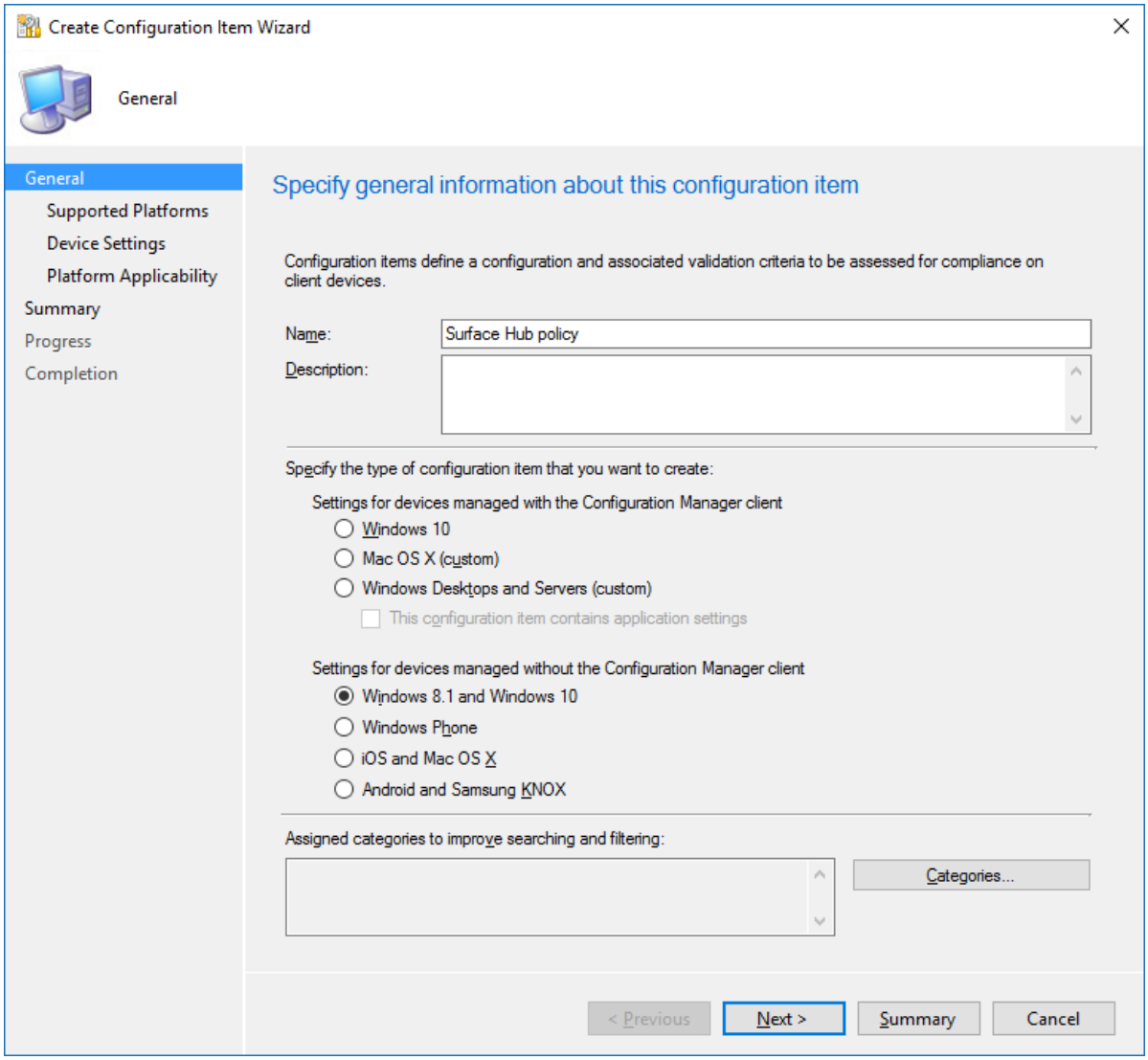
> **NOTE**
> These instructions are based on the current branch of System Center Configuration Manager.

To create a configuration item for Surface Hub settings

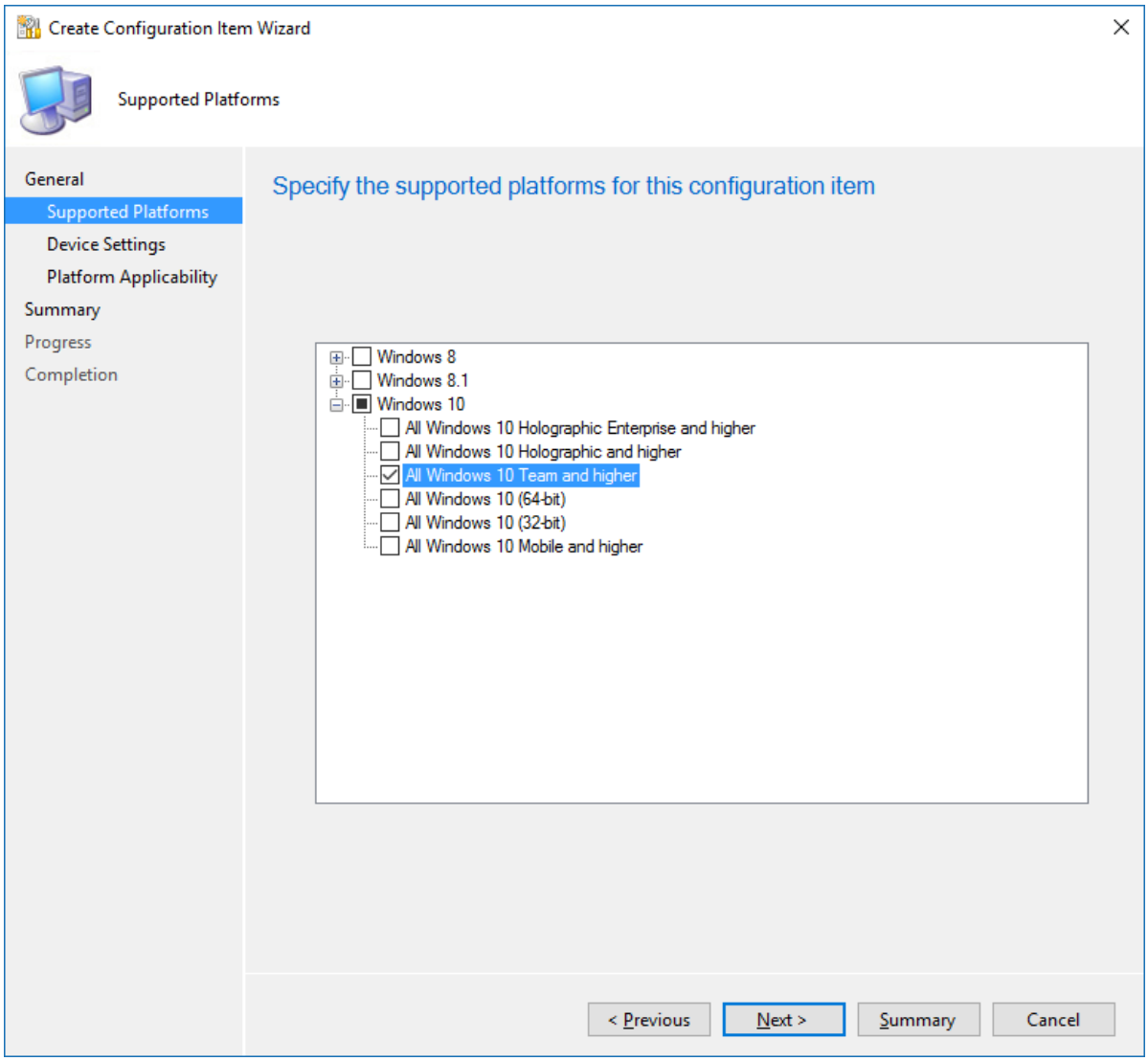
1. On the **Assets and Compliance** workspace of the Configuration Manager console, click **Overview** > **Compliance Settings** > **Configuration Items**.
2. On the **Home** tab, in the **Create** group, click **Create Configuration Item**.
3. On the **General** page of the Create Configuration Item Wizard, specify a name and optional description for the

configuration item.

4. Under **Settings for devices managed without the Configuration Manager client**, select **Windows 8.1 and Windows 10**, and then click **Next**.

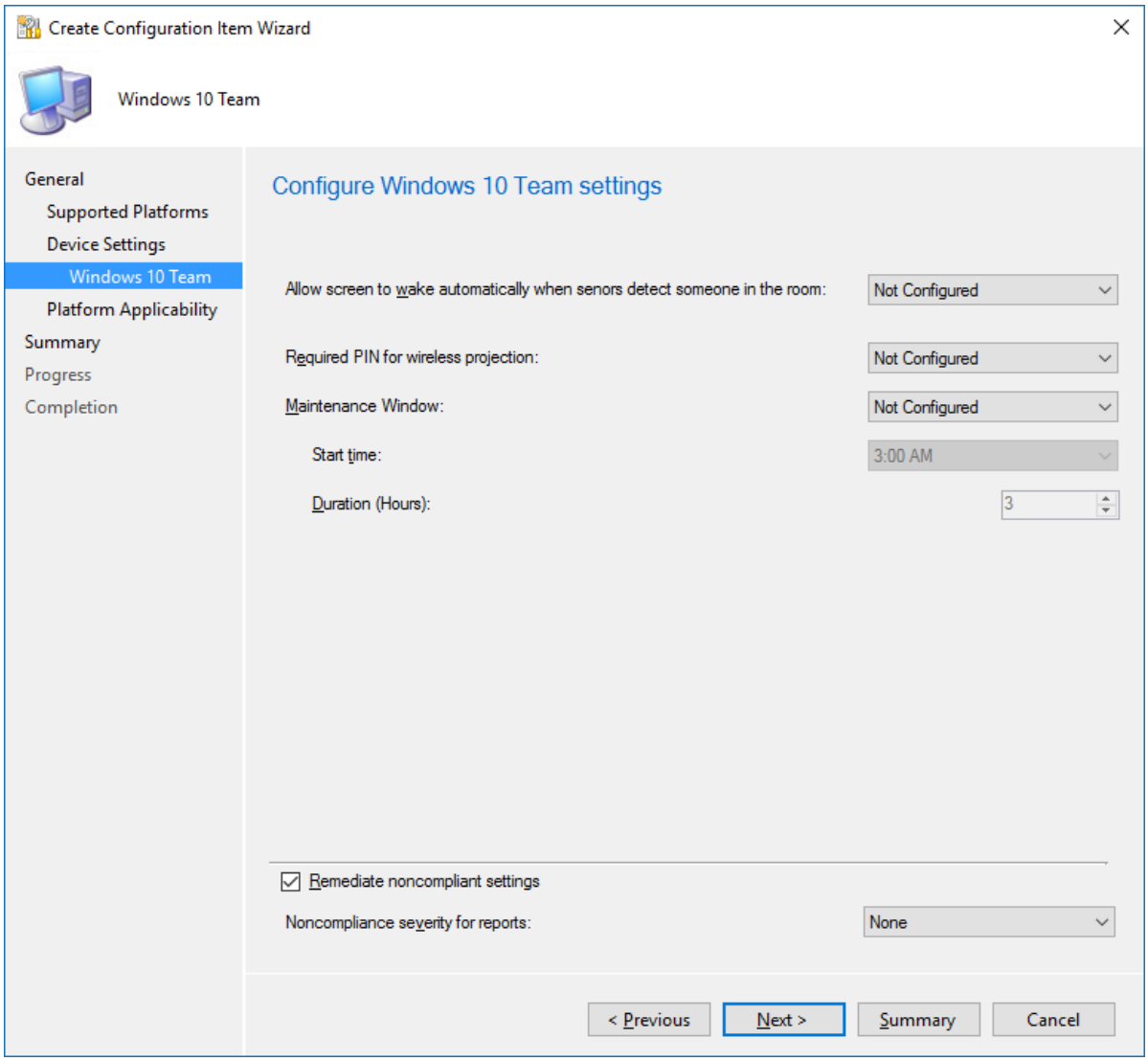


5. On the **Supported Platforms** page, expand **Windows 10** and select **All Windows 10 Team and higher**. Unselect the other Windows platforms, and then click **Next**.

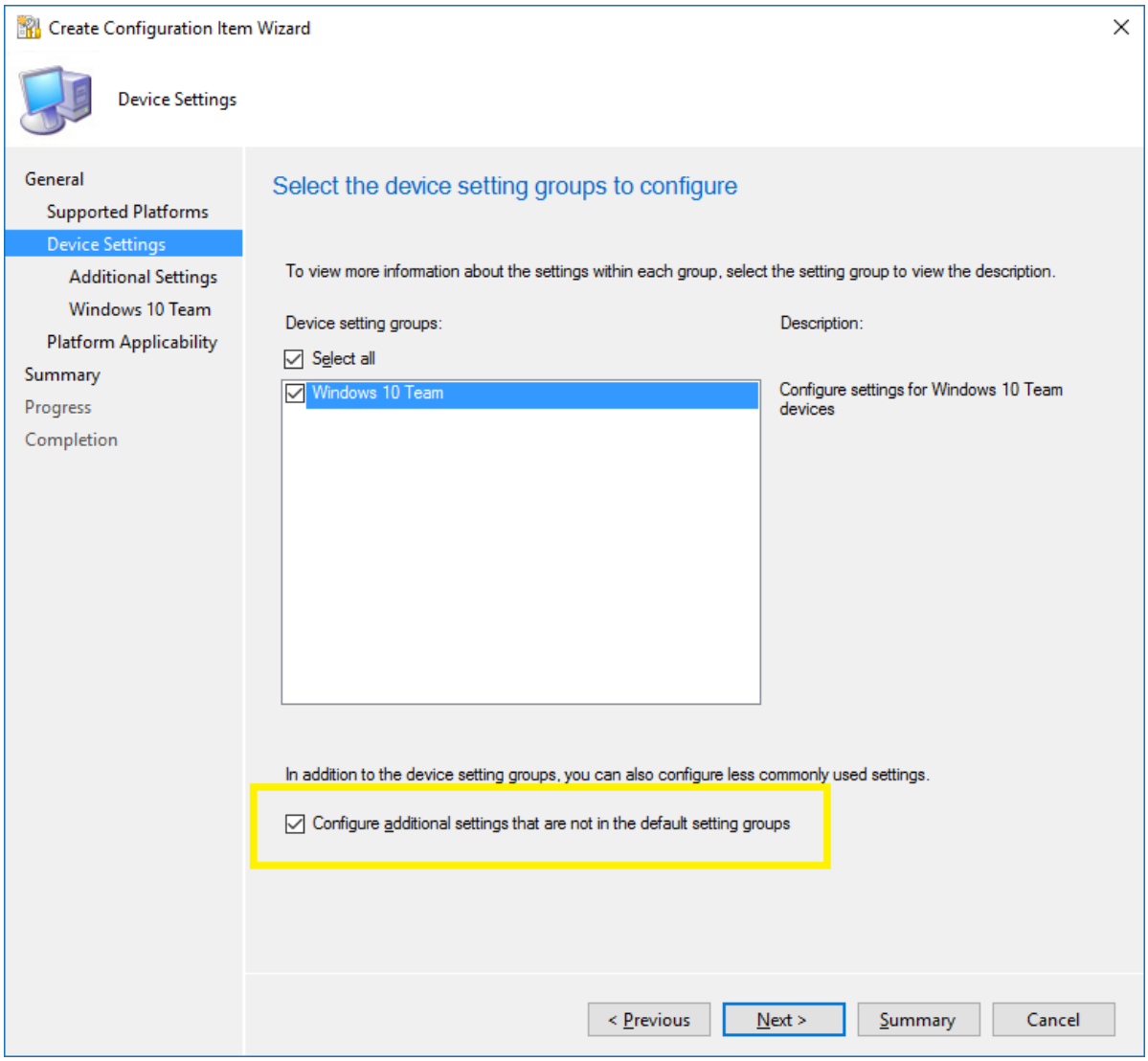


6. On the **Device Settings** page, under **Device settings groups**, select **Windows 10 Team**.

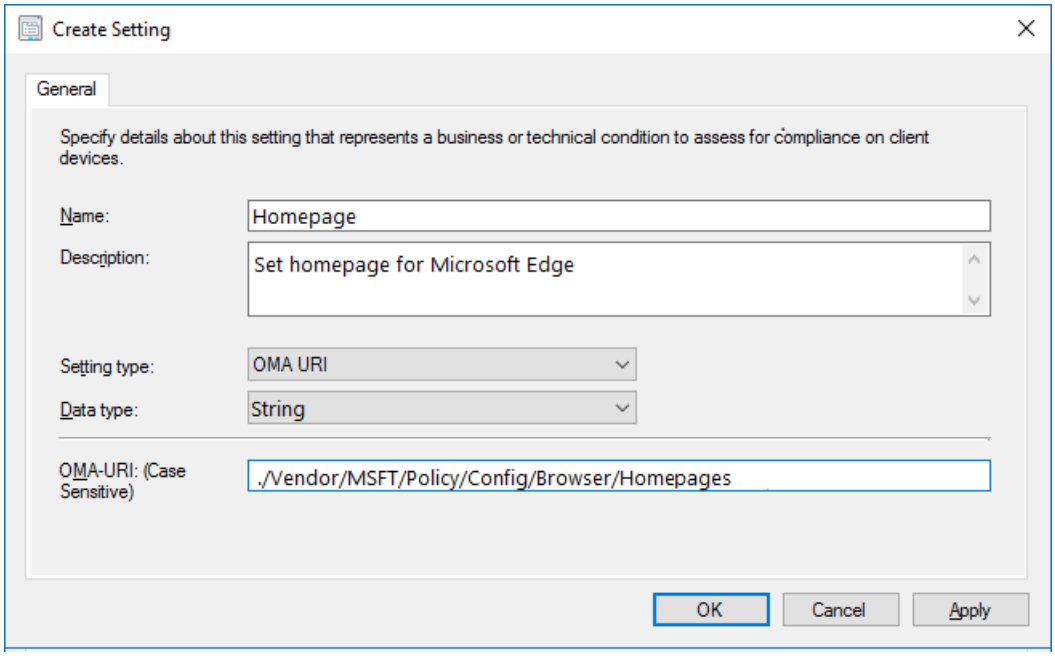
7. On the **Windows 10 Team** page, configure the settings you require.



8. You'll need to create custom settings to manage settings that are not available in the Windows 10 Team page. On the **Device Settings** page, select the check box **Configure additional settings that are not in the default setting groups**.



9. On the **Additional Settings** page, click **Add**.
10. In the **Browse Settings** dialog, click **Create Setting**.
11. In the **Create Setting** dialog, under the **General** tab, specify a name and optional description for the custom setting.
12. Under **Setting type**, select **OMA URI**.
13. Complete the form to create a new setting, and then click **OK**.



14. On the **Browse Settings** dialog, under **Available settings**, select the new setting you created, and then click **Select**.
15. On the **Create Rule** dialog, complete the form to specify a rule for the setting, and then click **OK**.
16. Repeat steps 9 to 15 for each custom setting you want to add to the configuration item.
17. When you're done, on the **Browse Settings** dialog, click **Close**.
18. Complete the wizard.
    You can view the new configuration item in the **Configuration Items** node of the **Assets and Compliance** workspace.

For more information, see Create configuration items for Windows 8.1 and Windows 10 devices managed without the System Center Configuration Manager client.

Related topics

Manage Microsoft Surface Hub

Microsoft Surface Hub administrator's guide

Monitor your Microsoft Surface Hub

5/4/2017 • 8 min to read • Edit Online

Monitoring for Microsoft Surface Hub devices is enabled through Microsoft Operations Management Suite (OMS). The Operations Management Suite is Microsoft's IT management solution that helps you manage and protect your entire IT infrastructure, including your Surface Hubs.

Surface Hub is offered as a Log Analytics solution in OMS, allowing you to collect and view usage and reliability data across all your Surface Hubs. Use the Surface Hub solution to:

- Inventory your Surface Hubs.
- View a snapshot of usage and reliability data for Skype meetings, wired and wireless projection, and apps on your Surface Hubs.
- Create custom alerts to respond quickly if your Surface Hubs report software or hardware issues.

Add Surface Hub to Operations Management Suite

1. **Sign in to Operations Management Suite (OMS)**. You can use either a Microsoft Account or a Work or School account to create a workspace. If your company is already using Azure Active Directory (Azure AD), use a Work or School account when you sign in to OMS. Using a Work or School account allows you to use identities from your Azure AD to manage permissions in OMS.
2. **Create a new OMS workspace**. Enter a name for the workspace, select the workspace region, and provide the email address that you want associated with this workspace. Select **Create**.
3. **Link Azure subscription to your workspace**. If your organization already has an Azure subscription, you can link it to your workspace. Note that you may need to request access from your organization's Azure administrator.

   > **NOTE**
   > If your organization does not have an Azure subscription, create a new one or select the default OMS Azure subscription from the list. Your workspace opens.

4. **Add Surface Hub solution**. In the Solutions Gallery, select the **Surface Hub** tile in the gallery and then select **Add** on the solution's details page. The solution is now visible on your workspace.

Use the Surface Hub dashboard

From the **Overview** page in your OMS workspace, click the Surface Hub tile to see the Surface Hub dashboard. Use the dashboard to get a snapshot of usage and reliability data across your Surface Hubs. Click into each view on the dashboard to see detailed data, modify the query as desired, and create alerts.

> **NOTE**
> Most of these views show data for the past 30 days, but this is subject to your subscription's data retention policy.

Active Surface Hubs

Use this view to get an inventory of all your Surface Hubs. Once connected to OMS, each Surface Hub periodically sends a "heartbeat" event to the server. This view shows Surface Hubs that have reported a heartbeat in the past

24 hours.

Wireless projection

Use this view to get usage and reliability data for wireless projection over the past 30 days. The graph shows the total number of wireless connections across all your Surface Hubs, which provides an indication whether people in your organization are using this feature. If it's a low number, it may suggest a need to provide training to help people in your organization learn how to wirelessly connect to a Surface Hub.

Also, the graph shows a breakdown of successful and unsuccessful connections. If you see a high number of unsuccessful connections, devices may not properly support wireless projection using Miracast. For best performance, Microsoft suggests that devices run a WDI Wi-Fi driver and a WDDM 2.0 graphics driver. Use the details view to learn if wireless projection problems are common with particular devices.

When a connection fails, users can also do the following if they are using a Windows laptop or phone:

- Remove the paired device from **Settings** > **Devices** > **Connected devices**, then try to connect again.
- Reboot the device.

Wired projection

Use this view to get usage and reliability data for wired projection over the past 30 days. If the graph shows a high number of unsuccessful connections, it may indicate a connectivity issue in your audio-visual pipeline. For example, if you use a HDMI repeater or a center-of-room control panel, they may need to be restarted.

Application usage

Use this view to get usage data for apps on your Surface Hubs over the past 30 days. The data comes from app launches on your Surface Hubs, not including Skype for Business. This view helps you understand which Surface Hub apps are the most valuable in your organization. If you are deploying new line-of-business apps in your environment, this can also help you understand how often they are being used.

Application Crashes

Use this view to get reliability data for apps on your Surface Hubs over the past 30 days. The data comes from app crashes on your Surface Hubs. This view helps you detect and notify app developers of poorly behaving in-box and line-of-business apps.

Sample Queries

Use this to create custom alerts based on a recommended set of queries. Alerts help you respond quickly if your Surface Hubs report software or hardware issues. For more inforamtion, see Set up alerts using sample queries.

Set up alerts with sample queries

Use alerts to respond quickly if your Surface Hubs report software or hardware issues. Alert rules automatically run log searches according to a schedule, and runs one or more actions if the results match specific criteria. For more information, see Alerts in Log Analytics.

The Surface Hub Log Analytics solution comes with a set of sample queries to help you set up the appropriate alerts and understand how to resolve issues you may encounter. Use them as a starting point to plan your monitoring and support strategy.

This table describes the sample queries in the Surface Hub solution:

ALERT TYPE	IMPACT	RECOMMENDED REMEDIATION	DETAILS

ALERT TYPE	IMPACT	RECOMMENDED REMEDIATION	DETAILS
Software	Error	**Reboot the device**. Reboot manually, or using the Reboot configuration service provider. Suggest doing this between meetings to minimize impact to your people in your organization.	Trigger conditions: - A critical process in the Surface Hub operating system, such as the shell, projection, or Skype, crashes or becomes non-responsive. - The device hasn't reported a heartbeat in the past 24 hours. This may be due to network connectivity issue or network-related hardware failure, or an error with the telemetry reporting system.
Software	Error	**Check your Exchange service**. Verify: - The service is available. - The device account password is up to date – see Password management for details.	Triggers when there's an error syncing the device calendar with Exchange.
Software	Error	**Check your Skype for Business service**. Verify: - The service is available. - The device account password is up to date – see Password management for details. - The domain name for Skype for Business is properly configured - see Configure a domain name.	Triggers when Skype fails to sign in.
Software	Error	**Reset the device**. This takes some time, so you should take the device offline. For more information, see Device reset.	Triggers when there is an error cleaning up user and app data at the end of a session. When this operation repeatedly fails, the device is locked to protect user data. You must reset the device to continue.
Hardware	Warning	**None**. Indicates negligible impact to functionality.	Triggers when there is an error with any of the following hardware components: - Virtual pen slots - NFC driver - USB hub driver - Bluetooth driver - Proximity sensor - Graphical performance (video card driver) - Mismatched hard drive - No keyboard/mouse detected

ALERT TYPE	IMPACT	RECOMMENDED REMEDIATION	DETAILS
Hardware	Error	**Contact Microsoft support**. Indicates impact to core functionality (such as Skype, projection, touch, and internet connectivity). **Note** Some events, including heartbeat, include the device's serial number that you can use when contacting support.	Triggers when there is an error with any of the following hardware components. **Components that affect Skype**: - Speaker driver - Microphone driver - Camera driver **Components that affect wired and wireless projection**: - Wired touchback driver - Wired ingest driver - Wireless adapter driver - Wi-Fi Direct error **Other components**: - Touch digitizer driver - Network adapter error (not reported to OMS)

To set up an alert

1. From the Surface Hub solution, select one of the sample queries.
2. Modify the query as desired. See Log Analytics search reference to learn more.
3. Click **Alert** at the top of the page to open the **Add Alert Rule** screen. See Alerts in Log Analytics for details on the options to configure the alert.
4. Click **Save** to complete the alert rule. It will start running immediately.

Enroll your Surface Hub

For Surface Hub to connect to and register with the OMS service, it must have access to the port number of your domains and the URLs. This table list the ports that OMS needs. For more information, see Configure proxy and firewall settings in Log Analytics.

> **NOTE**
> Surface Hub does not currently support the use of a proxy server to communicate with the OMS service.

AGENT RESOURCE	PORTS	BYPASS HTTPS INSPECTION?
*.ods.opinsights.azure.com	443	Yes
*.oms.opinsights.azure.com	443	Yes
*.blob.core.windows.net	443	Yes
ods.systemcenteradvisor.com	443	No

The Microsoft Monitoring Agent, used to connect devices to OMS, is integrated with the Surface Hub operating system, so there is no need to install additional clients to connect Surface Hub to OMS.

Once your OMS workspace is set up, there are several ways to enroll your Surface Hub devices:

- Settings app
- Provisioning package
- MDM provider, such as Microsoft Intune and Configuration Manager

You'll need the workspace ID and primary key of your OMS workspace. You can get these from the OMS portal.

Enroll using the Settings app

To Enroll using the settings app

1. From your Surface Hub, start **Settings**.
2. Enter the device admin credentials when prompted.
3. Select **This device**, and navigate to **Device management**.
4. Under **Monitoring**, select **Configure OMS settings**.
5. In the OMS settings dialog, select **Enable monitoring**.
6. Type the workspace ID and primary key of your OMS workspace. You can get these from the OMS portal.
7. Click **OK** to complete the configuration.

A confirmation dialog will appear telling you whether or not the OMS configuration was successfully applied to the device. If it was, the device will start sending data to OMS.

Enroll using a provisioning package

You can use a provisioning package to enroll your Surface Hub. For more infomation, see Create provisioning packages.

Enroll using a MDM provider

You can enroll Surface Hub into OMS using the SurfaceHub CSP. Intune and Configuration Manager provide built-in experiences to help create policy templates for Surface Hub. For more information, see Manage Surface Hub settings with an MDM provider.

Related topics

Manage Microsoft Surface Hub

Microsoft Surface Hub administrator's guide

Windows updates (Surface Hub)

5/4/2017 • 8 min to read • Edit Online

New releases of the Surface Hub operating system are published through Windows Update, just like releases of Windows 10. There are a couple of ways you can manage which updates are installed on your Surface Hubs, and the timing for when updates are applied.

- **Windows Update for Business** - New in Windows 10, Windows Update for Business is a set of features designed to provide enterprises additional control over how and when Windows Update installs releases, while reducing device management costs. Using this method, Surface Hubs are directly connected to Microsoft's Windows Update service.
- **Windows Server Update Services (WSUS)** - Set of services that enable IT administrators to obtain the updates that Windows Update determines are applicable to the devices in their enterprise, perform additional testing and evaluation on the updates, and select the updates they want to install. Using this method, Surface Hubs will receive updates from WSUS rather than Windows Update.

You can also configure Surface Hub to receive updates from both Windows Update for Business and WSUS. See Integrate Windows Update for Business with Windows Server Update Services for details.

CAPABILITIES	WINDOWS UPDATE FOR BUSINESS	WINDOWS SERVER UPDATE SERVICES (WSUS)
Receive updates directly from Microsoft's Windows Update service, with no additional infrastructure required.	Yes	No
Defer updates to provide additional time for testing and evaluation.	Yes	Yes
Deploy updates to select groups of devices.	Yes	Yes
Define maintenance windows for installing updates.	Yes	Yes

TIP

Use peer-to-peer content sharing to reduce bandwidth issues during updates. See Optimize update delivery for Windows 10 updates for details.

NOTE

Surface Hub does not currently support rolling back updates.

Surface Hub servicing model

Surface Hub uses the Windows 10 servicing model, referred to as Windows as a Service (WaaS). Traditionally, new features are added only in new versions of Windows that are released every few years. Each new version required lengthy and expensive processes to deploy in an organization. As a result, end users and organizations don't

frequently enjoy the benefits of new innovation. The goal of Windows as a Service is to continually provide new capabilities while maintaining a high level of quality.

Microsoft publishes two types of Surface Hub releases broadly on an ongoing basis:

- **Feature updates** - Updates that install the latest new features, experiences, and capabilities. Microsoft expects to publish an average of two to three new feature upgrades per year.
- **Quality updates** - Updates that focus on the installation of security fixes, drivers, and other servicing updates. Microsoft expects to publish one cumulative quality update per month.

In order to improve release quality and simplify deployments, all new releases that Microsoft publishes for Windows 10, including Surface Hub, will be cumulative. This means new feature updates and quality updates will contain the payloads of all previous releases (in an optimized form to reduce storage and networking requirements), and installing the release on a device will bring it completely up to date. Also, unlike earlier versions of Windows, you cannot install a subset of the contents of a Windows 10 quality update. For example, if a quality update contains fixes for three security vulnerabilities and one reliability issue, deploying the update will result in the installation of all four fixes.

The Surface Hub operating system is available on **Current Branch (CB)** and **Current Branch for Business (CBB)**. Like other editions of Windows 10, the servicing lifetime of CB or CBB is finite. You must install new feature updates on machines running these branches in order to continue receiving quality updates.

For more information on Windows as a Service, see Overview of Windows as a service.

Use Windows Update for Business

Surface Hubs, like all Windows 10 devices, include **Windows Update for Business (WUfB)** to enable you to control how your devices are being updated. Windows Update for Business helps reduce device management costs, provide controls over update deployment, offer quicker access to security updates, as well as provide access to the latest innovations from Microsoft on an ongoing basis. For more information, see Manage updates using Windows Update for Business.

To set up Windows Update for Business:

1. Group Surface Hub into deployment rings
2. Configure Surface Hub to use Current Branch or Current Branch for Business.
3. Configure when Surface Hub receives updates.

> **NOTE**
>
> You can use Microsoft Intune, System Center Configuration Manager, or a supported third-party MDM provider to set up WUfB. Walkthrough: use Microsoft Intune to configure Windows Update for Business.

Group Surface Hub into deployment rings

Use deployment rings to control when updates roll out to your Surface Hubs, giving you time to validate them. For example, you can update a small pool of devices first to verify quality before a broader roll-out to your organization. Depending on who manages Surface Hub in your organization, consider incorporating Surface Hub into the deployment rings that you've built for your other Windows 10 devices. For more information about deployment rings, see Build deployment rings for Windows 10 updates.

This table gives examples of deployment rings.

DEPLOYMENT RING	RING SIZE	SERVICING BRANCH	DEFERRAL FOR FEATURE UPDATES	DEFERRAL FOR QUALITY UPDATES (SECURITY FIXES, DRIVERS, AND OTHER UPDATES)	VALIDATION STEP
Evaluation (e.g. non-critical or test devices)	Small	Current Branch (CB)	None. Devices receive feature updates immediately after CB is released.	None. Devices receive quality updates immediately after CB is released.	Manually test and evaluate new functionality. Pause updates if there are issues.
Pilot (e.g. devices used by select teams)	Medium	Current Branch for Business (CBB)	None. Devices receive feature updates immediately once CBB is released.	None. Devices receive quality updates immediately after CBB is released.	Monitor device usage and user feedback. Pause updates if there are issues.
Broad deployment (e.g. most of the devices in your organization)	Large	Current Branch for Business (CBB)	60 days after CBB is released.	14 days after CBB is released.	Monitor device usage and user feedback. Pause updates if there are issues.
Mission critical (e.g. devices in executive boardrooms)	Small	Current Branch for Business (CBB)	180 days after CBB is released (maximum deferral for feature updates).	30 days after CBB is released (maximum deferral for quality updates).	Monitor device usage and user feedback.

Configure Surface Hub to use Current Branch or Current Branch for Business

By default, Surface Hubs are configured to receive updates from Current Branch (CB). CB receives feature updates as soon as they are released by Microsoft. Current Branch for Business (CBB), on the other hand, receives feature updates at least four months after they have been initially offered to CB devices, and includes all of the quality updates that have been released in the interim. For more information on the differences between CB and CBB, see Servicing branches.

To manually configure Surface Hub to use CB or CBB:

1. Open **Settings** > **Update & Security** > **Windows Update**, and then select **Advanced Options**.
2. Select **Defer feature updates**.

To configure Surface Hub to use CB or CBB remotely using MDM, set an appropriate Update/BranchReadinessLevel policy.

Configure when Surface Hub receives updates

Once you've determined deployment rings for your Surface Hubs, configure update deferral policies for each ring:

- To defer feature updates, set an appropriate Update/DeferFeatureUpdatesPeriodInDays policy for each ring.
- To defer quality updates, set an appropriate Update/DeferQualityUpdatesPeriodInDays policy for each ring.

> **NOTE**
>
> If you encounter issues during the update rollout, you can pause updates using Update/PauseFeatureUpdates and Update/PauseQualityUpdates.

Use Windows Server Update Services

You can connect Surface Hub to your Windows Server Update Services (WSUS) server to manage updates. Updates will be controlled through approvals or automatic deployment rules configured in your WSUS server, so new upgrades will not be deployed until you choose to deploy them.

To manually connect a Surface Hub to a WSUS server:

1. Open **Settings** on your Surface Hub.
2. Enter the device admin credentials when prompted.
3. Navigate to **Update & security** > **Windows Update** > **Advanced options** > **Configure Windows Server Update Services (WSUS) server**.
4. Click **Use WSUS Server to download updates** and type the URL of your WSUS server.

To connect Surface Hub to a WSUS server using MDM, set an appropriate Update/UpdateServiceUrl policy.

If you use a proxy server or other method to block URLs

If you use a method other than WSUS to block specific URLs and prevent updates, you will need to add the following Windows update trusted site URLs to the "allow list":

- `http(s)://*.update.microsoft.com`
- `http://download.windowsupdate.com`
- `http://windowsupdate.microsoft.com`

Once the Windows 10 Team Anniversary Update is installed, you can remove these addresses to return your Surface Hub to its previous state.

Maintenance window

To ensure the device is always available for use during business hours, Surface Hub performs its administrative functions during a specified maintenance window. During the maintenance window, the Surface Hub automatically installs updates through Windows Update or WSUS, and reboots the device if needed.

Surface Hub follows these guidelines to apply updates:

- Install the update during the next maintenance window. If a meeting is scheduled to start during a maintenance window, or the Surface Hub sensors detect that the device is being used, the pending update will be postponed to the following maintenance window.
- If the next maintenance window is past the update's prescribed grace period, the device will calculate the next available slot during business hours using the estimated install time from the update's metadata. It will continue to postpone the update if a meeting is scheduled, or the Surface Hub sensors detect that the device is being used.
- If a pending update is past the update's prescribed grace period, the update will be immediately installed. If a reboot is needed, the Surface Hub will automatically reboot during the next maintenance window.

> **NOTE**
> Allow time for updates when you first setup your Surface Hub. For example, a backlog of virus definitions may be available, which should be immediately installed.

A default maintenance window is set for all new Surface Hubs:

- **Start time:** 3:00 AM
- **Duration:** 1 hour

To manually change the maintenance window:

1. Open **Settings** on your Surface Hub.
2. Navigate to **Update & security** > **Windows Update** > **Advanced options**.
3. Under **Maintenance hours**, select **Change**.

To change the maintenance window using MDM, set the **MOMAgent** node in the SurfaceHub configuration service provider. See Manage settings with an MDM provider for more details.

Related topics

Manage Microsoft Surface Hub

Microsoft Surface Hub administrator's guide

Manage Surface Hub settings

5/4/2017 • 1 min to read • Edit Online

In this section

TOPIC	DESCRIPTION
Local management for Surface Hub settings	Learn about Surface Hub settings.
Accessibility	Accessibility settings for the Surface Hub can be changed by using the Settings app. You'll find them under Ease of Access. Your Surface Hub has the same accessibility options as Windows 10.
Change the Surface Hub device account	You can change the device account in Settings to either add an account if one was not already provisioned, or to change any properties of an account that was already provisioned.
Device reset	You may need to reset your Surface Hub.
Use fully qualified domain name with Surface Hub	Options to configure domain name with Surface Hub.
Wireless network management	Surface Hub offers two options for network connectivity to your corporate network and Internet: wireless, and wired. While both provide network access, we recommend you use a wired connection.

Local management for Surface Hub settings

5/4/2017 • 2 min to read • Edit Online

After initial setup of Microsoft Surface Hub, the device's settings can be locally managed through **Settings**.

Surface Hub settings

Surface Hubs have many settings that are common to other Windows devices, but also have settings which are only configurable on Surface Hubs. This table lists settings only cofigurable on Surface Hubs.

SETTING	LOCATION	DESCRIPTION
Device account	This device > Accounts	Set or change the Surface Hub's device account.
Device account sync status	This device > Accounts	Check the sync status of the device account's mail and calendar on the Surface Hub.
Password rotation	This device > Accounts	Choose whether to let the Surface Hub automatically rotate the device account's password.
Change admin account password	This device > Accounts	Change the password for the local admin account. This is only available if you configured the device to use a local admin during first run.
Configure Operations Management Suite (OMS)	This device > Device management	Set up monitoring for your Surface Hub using OMS.
Open the Microsoft Store app	This device > Apps & features	The Microsoft Store app is only available to admins through the Settings app.
Skype for Business domain name	This device > Calling	Configure a domain name for your Skype for Business server.
Default microphone and speaker settings	This device > Calling	Configure a default microphone and speaker for calls, and a default speaker for media playback.
Turn off wireless projection using Miracast	This device > Wireless projection	Choose whether presenters can wirelessly project to the Surface Hub using Miracast.
Require a PIN for wireless projection	This device > Wireless projection	Choose whether people are required to enter a PIN before they use wireless projection.
Wireless projection (Miracast) channel	This device > Wireless projection	Set the channel for Miracast projection.

SETTING	LOCATION	DESCRIPTION
Meeting info shown on the welcome screen	This device > Welcome screen	Choose whether meeting organizer, time, and subject show up on the welcome screen.
Welcome screen background	This device > Welcome screen	Choose a background image for the welcome screen.
Turn on screen with motion sensors	This device > Session & clean up	Choose whether the screen turns on when motion is detected.
Session time out	This device > Session & clean up	Choose how long the device needs to be inactive before returning to the welcome screen.
Sleep time out	This device > Session & clean up	Choose how long the device needs to be inactive before going to sleep mode.
Friendly name	This device > About	Set the Surface Hub name that people will see when connecting wirelessly.
Maintenance hours	Update & security > Windows Update > Advanced options	Configure when updates can be installed.
Configure Windows Server Update Services (WSUS) server	Update & security > Windows Update > Advanced options	Change whether Surface Hub receives updates from a WSUS server instead of Windows Update.
Save BitLocker key	Update & security > Recovery	Backup your Surface Hub's BitLocker key to a USB drive.
Collect logs	Update & security > Recovery	Save logs to a USB drive to send to Microsoft later.

Related topics

Manage Surface Hub settings

Remote Surface Hub management

Microsoft Surface Hub administrator's guide

Accessibility (Surface Hub)

5/4/2017 • 1 min to read • Edit Online

Microsoft Surface Hub has the same accessibility options as Windows 10.

Default accessibility settings

The full list of accessibility settings are available to IT admins in the **Settings** app. The default accessibility settings for Surface Hub include:

ACCESSIBILITY FEATURE	DEFAULT SETTINGS
Narrator	Off
Magnifier	Off
High contrast	No theme selected
Closed captions	Defaults selected for Font and Background and window
Keyboard	**On-screen Keyboard**, **Sticky Keys**, **Toggle Keys**, and **Filter Keys** are all off.
Mouse	Defaults selected for **Pointer size**, **Pointer color** and **Mouse keys**.
Other options	Defaults selected for **Visual options** and **Touch feedback**.

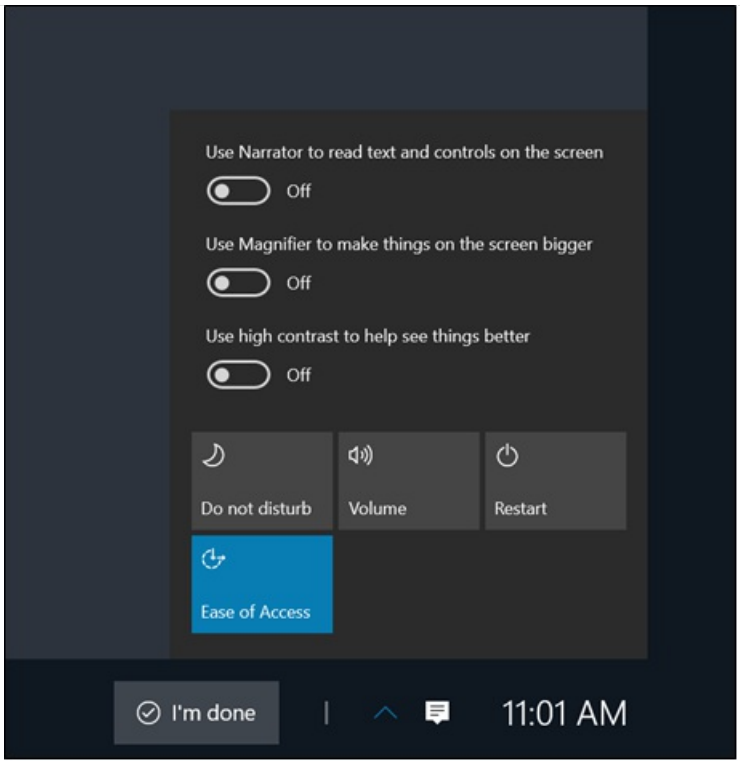
Additionally, these accessibility features and apps are returned to default settings when users press I'm Done:

- Narrator
- Magnifier
- High contrast
- Filter keys
- Sticky keys
- Toggle keys
- Mouse keys

Change accessibility settings during a meeting

During a meeting, users can toggle accessibility features and apps in a couple ways:

- Keyboard shortcuts
- **Quick Actions** > **Ease of Access** from the status bar



Related topics

Manage Microsoft Surface Hub

Microsoft Surface Hub administrator's guide

Change the Microsoft Surface Hub device account

5/4/2017 • 1 min to read • Edit Online

You can change the device account in Settings to either add an account if one was not already provisioned, or to change any properties of an account that was already provisioned.

Details

VALUE	DESCRIPTION
User Principal Name	The user principal name (UPN) of the device account.
Password	The corresponding password of the device account.
Domain	The domain that the device account belongs to. This field does not need to be provided for Office 365 accounts.
User name	The user name of the device account. This field does not need to be provided for Office 365 accounts.
Session Initiation Protocol (SIP) address	The SIP address of the device account.
Microsoft Exchange server	This is the Exchange server of the device account. The device account's username and password must be able to authenticate to the specified Exchange server.
Enable Exchange services	When checked, all Exchange services will be enabled (for example, calendar on the welcome screen, emailing whiteboards). When not checked, all Exchange services will be disabled, and the Exchange server does not need to be provided.

What happens?

The UPN and password are used to validate the account in AD or Azure AD. If the validation fails, you may need to provide the domain and user name.

Using the credentials provided, we will try to discover the SIP address. If a SIP address can't be found, then Skype for Business will use the UPN as the SIP address. If this is not the SIP address for the account, you will need to provide the SIP address.

The Exchange server address will need to be provided if the device can't find a server associated with the login credentials. Microsoft Surface Hub will use the Exchange server to talk to ActiveSync, which enables several key features on the device.

Related topics

Manage Microsoft Surface Hub

Microsoft Surface Hub administrator's guide

Device reset (Surface Hub)

5/4/2017 • 1 min to read • Edit Online

You may wish to reset your Microsoft Surface Hub.

Typical reasons for a reset include:

- The device isn't running well after installing an update.
- You're repurposing the device for a new meeting space and want to reconfigure it.
- You want to change how you locally manage the device.

Initiating a reset will return the device to the last cumulative Windows update, and remove all local user files and configuration, including:

- The device account
- MDM enrollment
- Domain join or Azure AD join information
- Local admins on the device
- Configurations from MDM or the Settings app

> **IMPORTANT**
> Performing a device reset may take up to 6 hours. Do not turn off or unplug the Surface Hub until the process has completed. Interrupting the process will render the device inoperable, requiring warranty service to return to normal functionality.

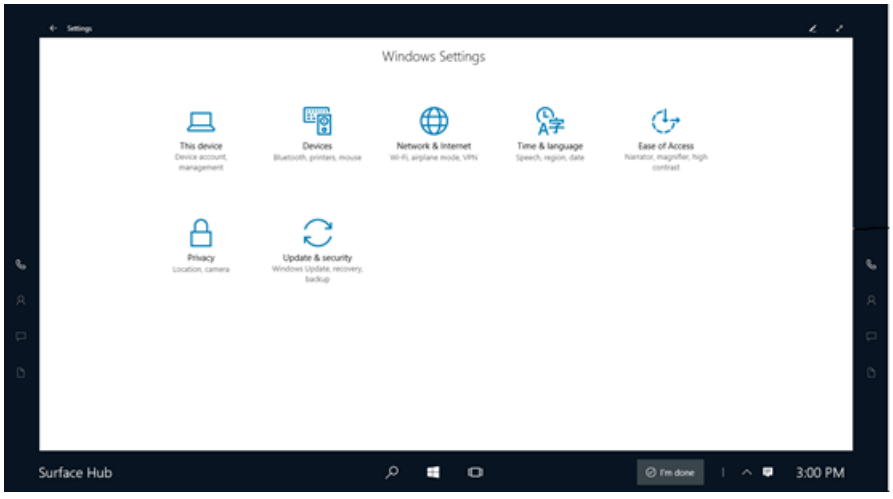
After the reset, Surface Hub restarts the first run program again. If the Surface Hub displays a Welcome screen, that indicates that the reset encountered a problem and rolled back to the previously existing OS image.

If you see a blank screen for long periods of time during the **Reset device** process, please wait and do not take any action.

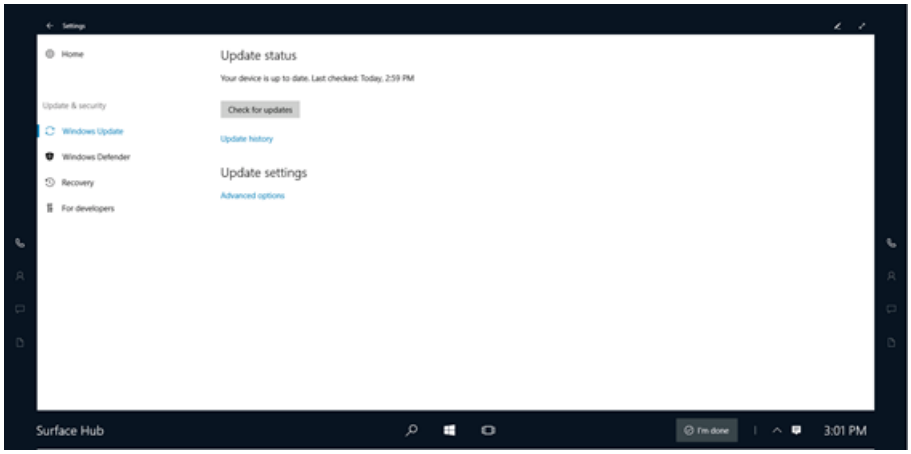
Reset a Surface Hub from Settings

To reset a Surface Hub

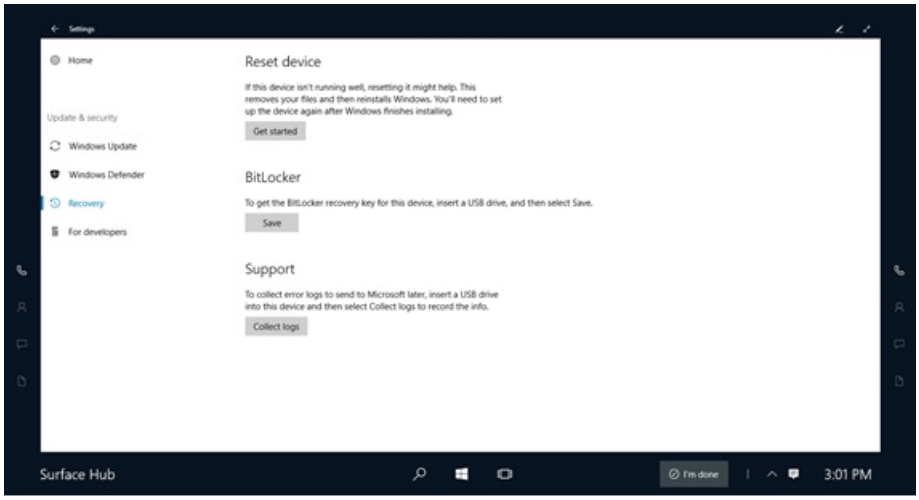
1. On your Surface Hub, open **Settings**.



2. Click **Update & Security**.



3. Click **Recovery**, and then click **Get started**.



Reset a Surface Hub from Windows Recovery Environment

On rare occasions, a Surface Hub may encounter an error while cleaning up user and app data at the end of a session. When this happens, the device will automatically reboot and try again. But if this operation fails repeatedly, the device will be automatically locked to protect user data. To unlock it, you must reset the device from Windows Recovery Environment (Windows RE).

To reset a Surface Hub from Windows Recovery Environment

1. From the welcome screen, toggle the Surface Hub's power switch 3 times. Wait a few seconds between each toggle. See the Surface Hub Site Readiness Guide for help with locating the power switch.
2. The device should automatically boot into Windows RE. Select **Advanced Repair**.
3. Select **Reset**.
4. If prompted, enter your device's BitLocker key.

Related topics

Manage Microsoft Surface Hub

Microsoft Surface Hub administrator's guide

Configure domain name for Skype for Business

5/4/2017 • 1 min to read • Edit Online

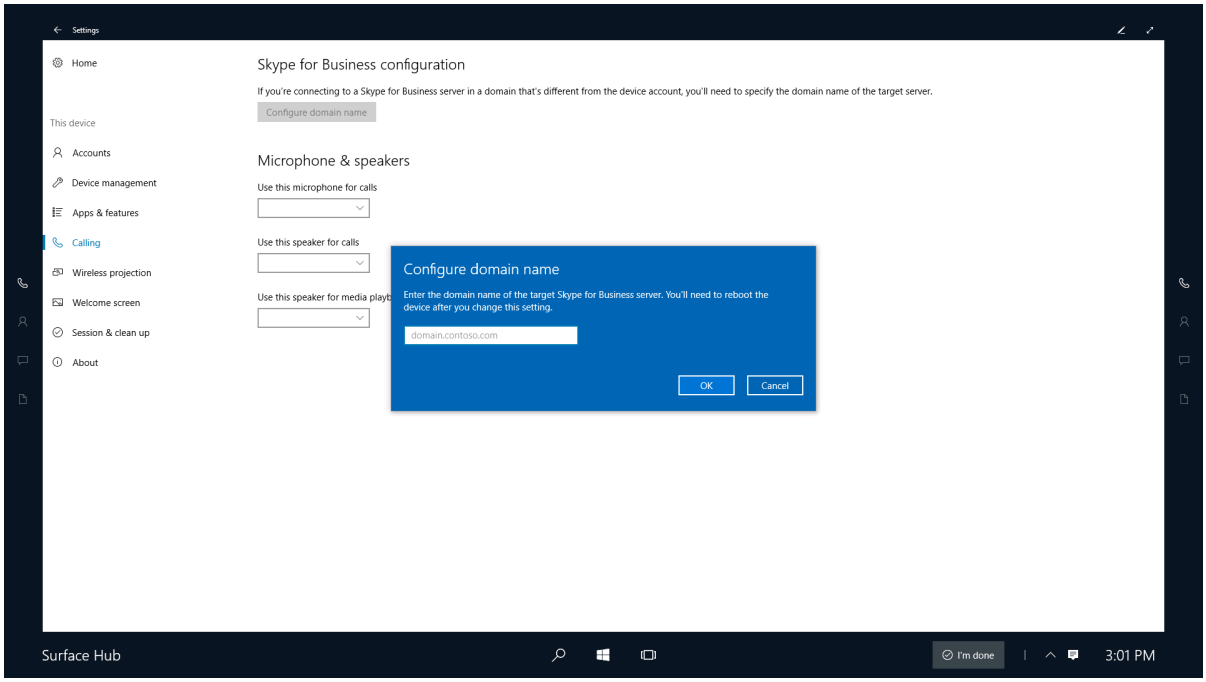
There are a few scenarios where you need to specify the domain name of your Skype for Business server:

- **Multiple DNS suffixes** - When your Skype for Business infrastructure has disjointed namespaces such that one or more servers have a DNS suffix that doesn't match the suffix of the sign-in address (SIP) for Skype for Business.
- **Skype for Business and Exchange suffixes are different** - When the suffix of the sign-in address for Skype for Business differs from the suffix of the Exchange address used for the device account.
- **Working with certificates** - Large organizations with on-premise Skype for Business servers commonly use certificates with their own root certificate authority (CA). It is common for the CA domain to be different than the domain of the Skype for Business server which causes the certificate to not be trusted, and sign-in fails. Skype needs to know the domain name of the certificate in order to set up a trust relationship. Enterprises typically use Group Policy to push this out to Skype desktop, but Group Policy is not supported on Surface Hub.

To configure the domain name for your Skype for Business server

1. On Surface Hub, open **Settings**.
2. Click **This device**, and then click **Calling**.
3. Under **Skype for Business configuration**, click **Configure domain name**.
4. Type the domain name for your Skype for Business server, and then click **Ok**.

> **TIP**
>
> You can type multiple domain names, separated by commas.
> For example: lync.com, outlook.com, lync.glbdns.microsoft.com



Wireless network management (Surface Hub)

5/4/2017 • 1 min to read • Edit Online

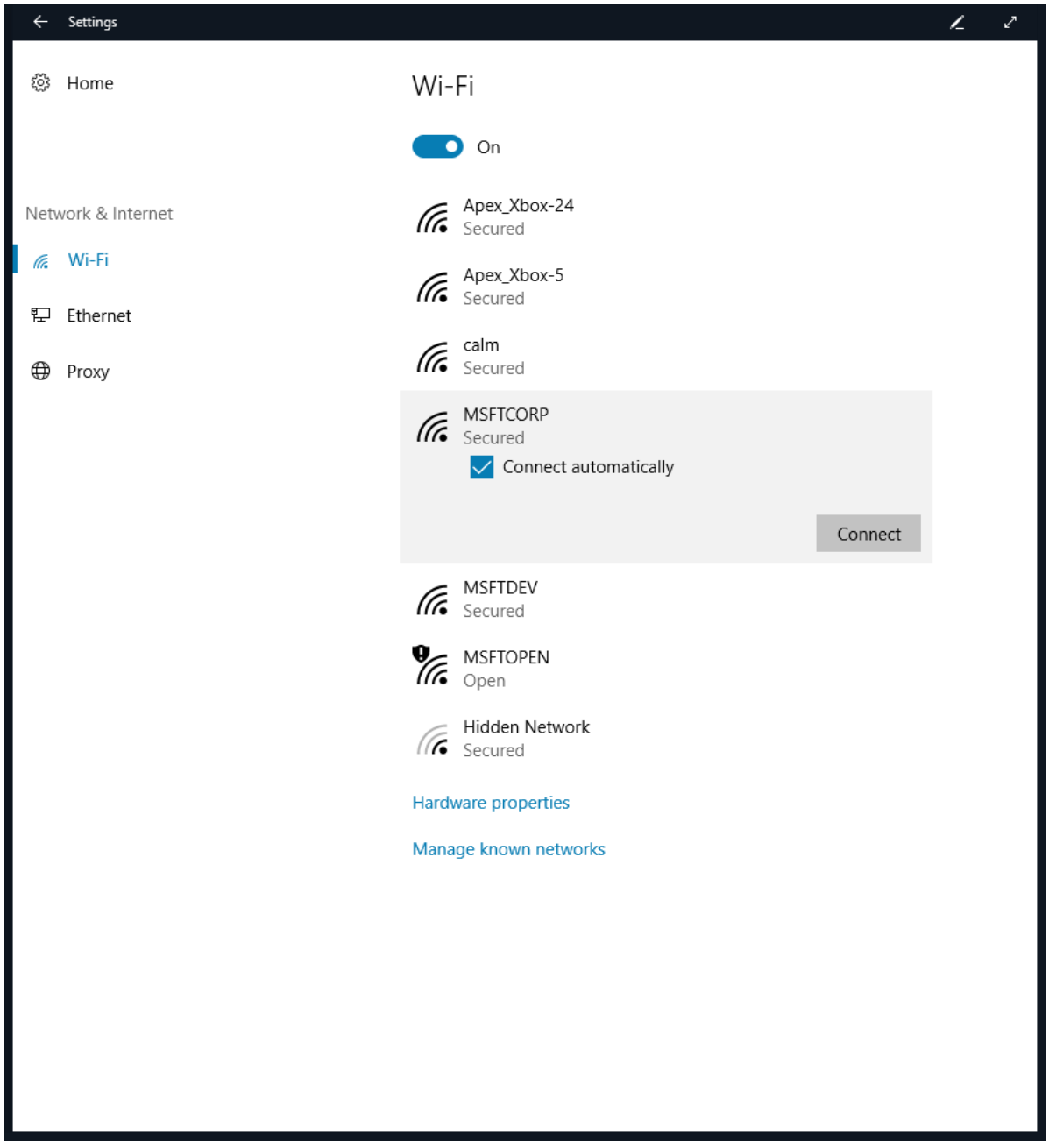
Microsoft Surface Hub offers two options for network connectivity to your corporate network and Internet: wireless, and wired. While both provide network access, we recommend you use a wired connection.

Modifying, adding, or reviewing a network connection

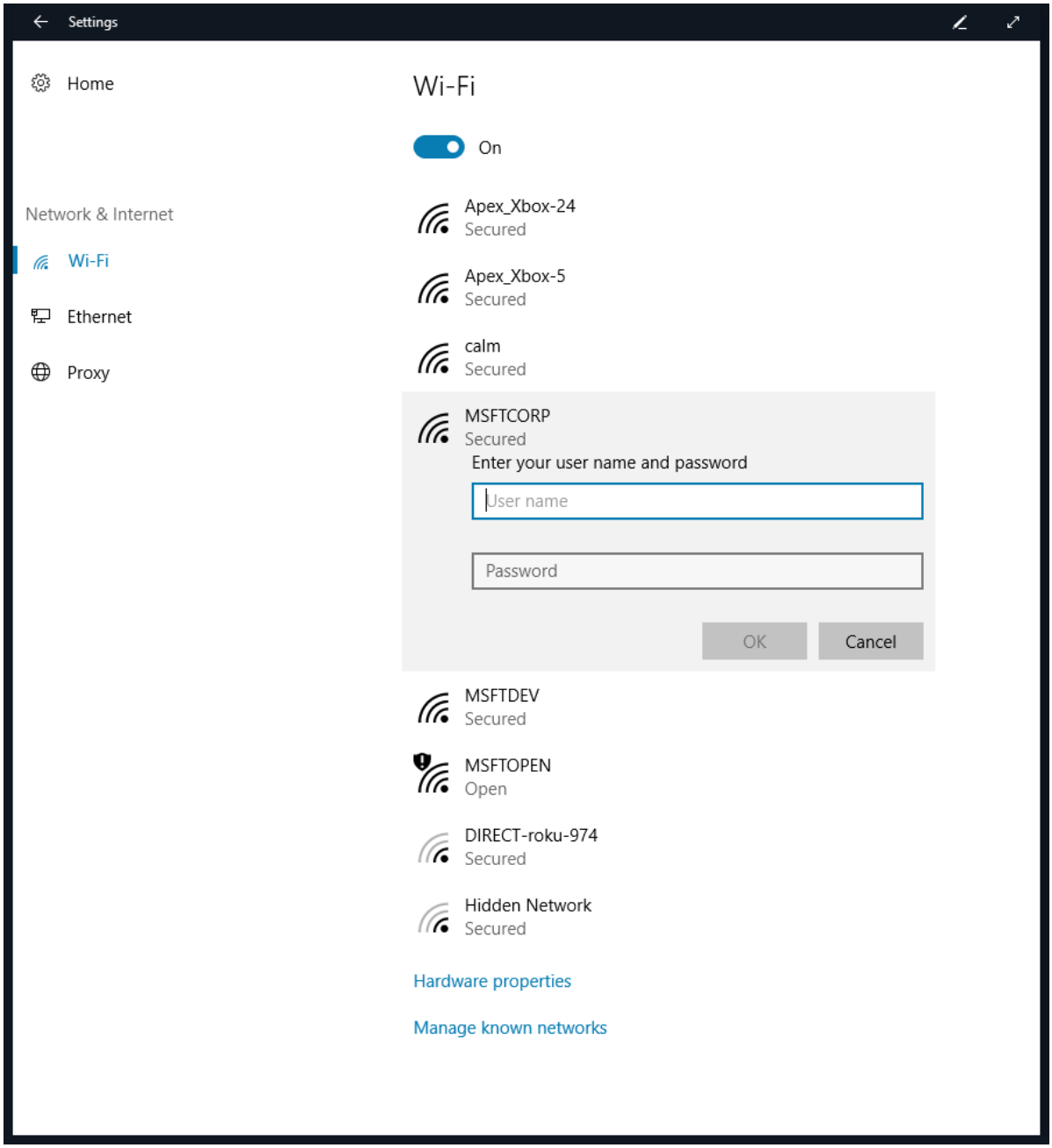
If a wired network connection is not available, the Surface Hub can use a wireless network for internet access. A properly connected and configured Wi-Fi access point must be available and within range of the Surface Hub.

Choose a wireless access point

1. On the Surface Hub, open **Settings** and enter your admin credentials.
2. Click **System**, and then click **Network & Internet**. Under **Wi-Fi**, choose an access point. If you want Surface Hub to automatically connect to this access point, click **Connect automatically**. Click **Connect**.

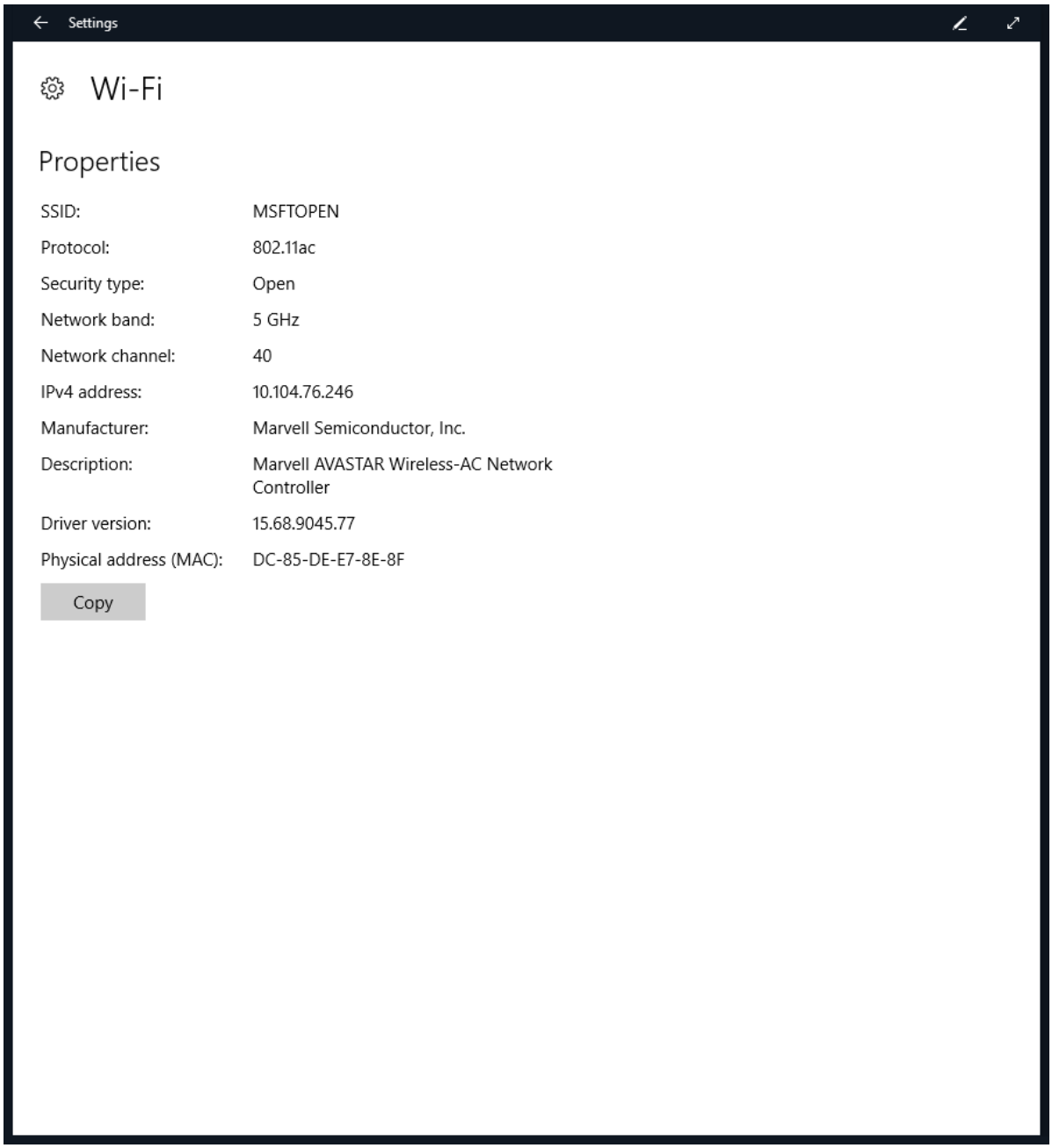


3. If the network is secured, you'll be asked to enter the security key. Click **Next** to connect.



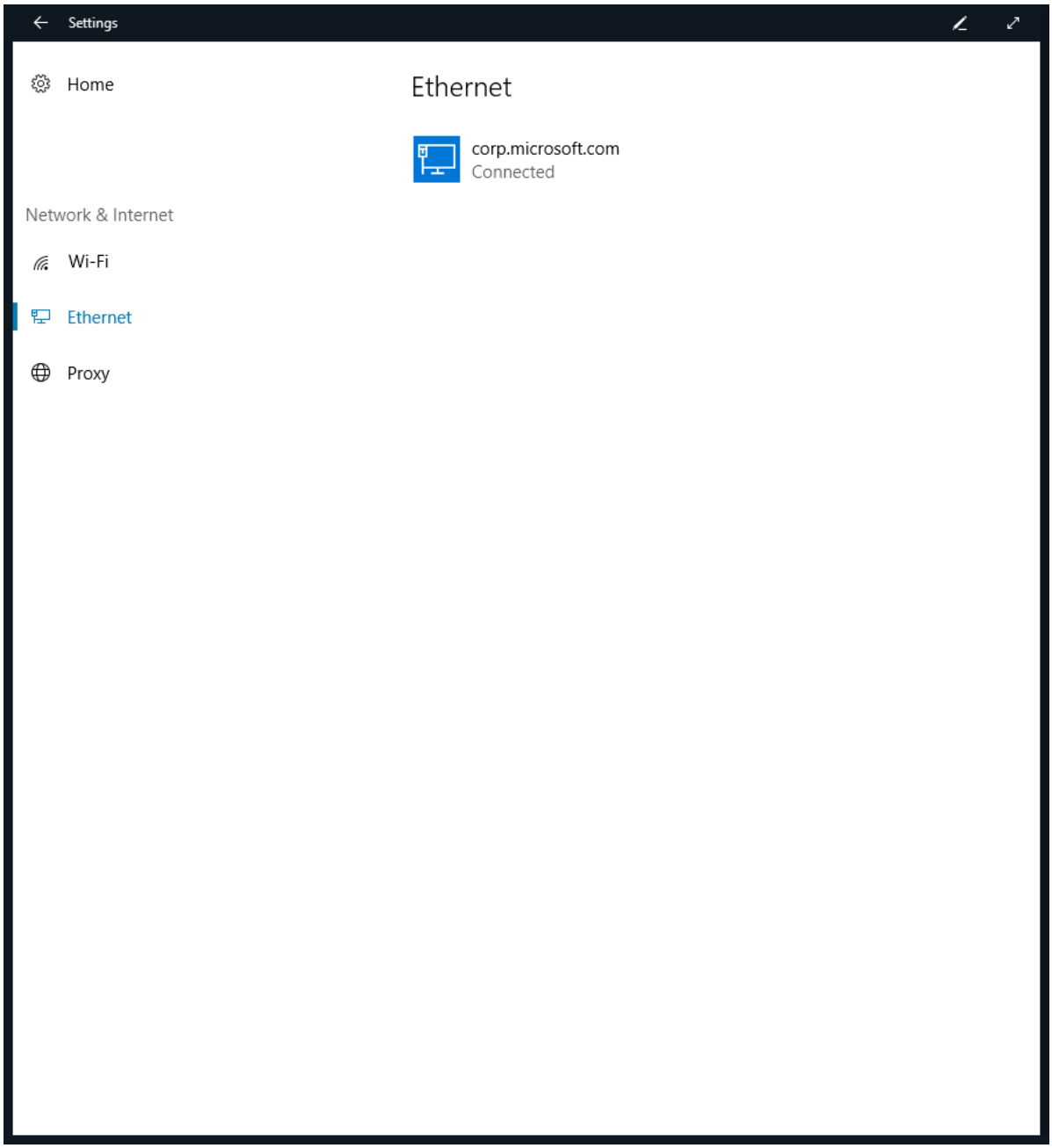
Review wireless settings

1. On the Surface Hub, open **Settings** and enter your admin credentials.
2. Click **System**, click **Network & Internet**, then **Wi-Fi**, and then click **Advanced options**.
3. Surface Hub shows you the properties for the wireless network connection.

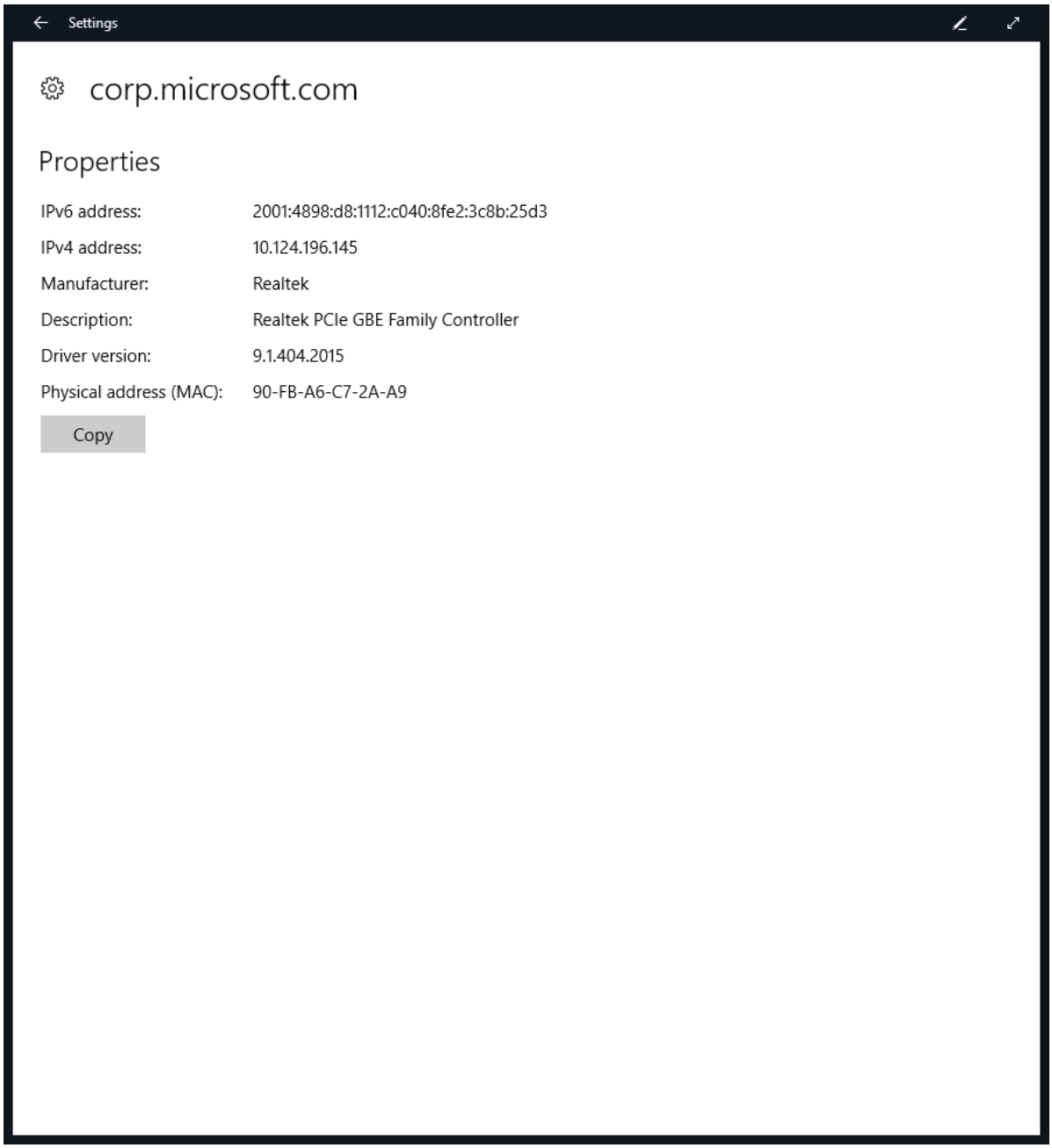


Review wired settings

1. On the Surface Hub, open **Settings** and enter your admin credentials.
2. Click **System**, click **Network & Internet**, then click on the network under Ethernet.



3. The system will show you the properties for the wired network connection.



Related topics

Manage Microsoft Surface Hub

Microsoft Surface Hub administrator's guide

Install apps on your Microsoft Surface Hub

5/4/2017 • 7 min to read • Edit Online

You can install additional apps on your Surface Hub to fit your team or organization's needs. There are different methods for installing apps depending on whether you are developing and testing an app, or deploying a released app. This topic describes methods for installing apps for either scenario.

A few things to know about apps on Surface Hub:

- Surface Hub only runs Universal Windows Platform (UWP) apps. See a list of apps that work with Surface Hub.
- Apps must be targeted for the Universal device family.
- By default, apps must be Store-signed to be installed. During testing and development, you can also choose to run developer-signed UWP apps by placing the device in developer mode.- When submitting an app to the Microsoft Store, developers need to set Device family availability and Organizational licensing options to make sure an app will be available to run on Surface Hub.
- You need admin credentials to install apps on your Surface Hub. Since the device is designed to be used in communal spaces like meeting rooms, people can't access the Microsoft Store to download and install apps.

Develop and test apps

While you're developing your own app, there are a few options for testing apps on Surface Hub.

Developer Mode

By default, Surface Hub only runs UWP apps that have been published to and signed by the Microsoft Store. Apps submitted to the Microsoft Store go through security and compliance tests as part of the app certification process, so this helps safeguard your Surface Hub against malicious apps.

By enabling developer mode, you can also install developer-signed UWP apps.

> **IMPORTANT**
> After developer mode has been enabled, you will need to reset the Surface Hub to disable it. Resetting the device removes all local user files and configurations and then reinstalls Windows.

To turn on developer mode

1. From your Surface Hub, start **Settings**.
2. Type the device admin credentials when prompted.
3. Navigate to **Update & security** > **For developers**.
4. Select **Developer mode** and accept the warning prompt.

Visual Studio

During development, the easiest way to test your app on a Surface Hub is using Visual Studio. Visual Studio's remote debugging feature helps you discover issues in your app before deploying it broadly. For more information, see Test Surface Hub apps using Visual Studio.

Provisioning package

Use Visual Studio to create an app package for your UWP app, signed using a test certificate. Then use Windows Imaging and Configuration Designer (ICD) to create a provisioning package containing the app package. For more information, see Create provisioning packages.

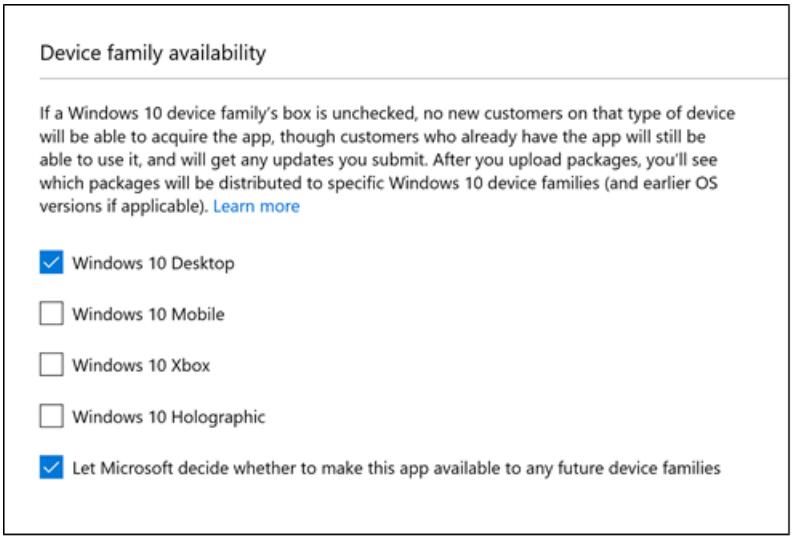
Submit apps to the Microsoft Store

Once an app is ready for release, developers need to submit and publish it to the Microsoft Store. For more information, see Publish Windows apps.

During app submission, developers need to set **Device family availability** and **Organizational licensing** options to make sure the app will be available to run on Surface Hub.

To set device family availability

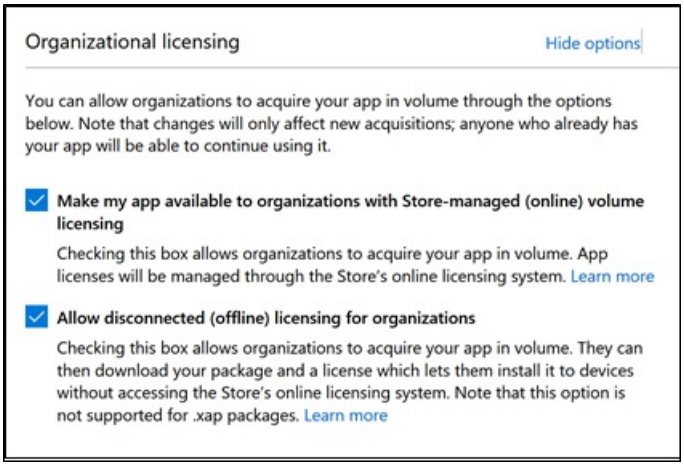
1. On the Windows Dev Center, navigate to your app submission page.
2. Select **Packages**.
3. Under Device family availability, select these options:
   - **Windows 10 Desktop** (other device families are optional)
   - **Let Microsoft decide whether to make the app available to any future device families**



For more information, see Device family availability.

To set organizational licensing

1. On the Windows Dev Center, navigate to your app submission page.
2. Select **Pricing and availability**.
3. Under Organizational licensing, select **Allow disconnected (offline) licensing for organizations**.



> **NOTE**
>
> **Make my app available to organizations with Store-managed (online) licensing and distribution** is selected by default.

> **NOTE**
>
> Developers can also publish line-of-business apps directly to enterprises without making them broadly available in the Store. For more information, see Distribute LOB apps to enterprises.

For more information, see Organizational licensing options.

Deploy released apps

There are several options for installing apps that have been released to the Microsoft Store, depending on whether you want to evaluate them on a few devices, or deploy them broadly to your organization.

To install released apps:

- Download the app using the Microsoft Store app, or
- Download the app package from the Microsoft Store for Business, and distribute it using a provisioning package or a supported MDM provider.

Microsoft Store app

To evaluate apps released on the Microsoft Store, use the Microsoft Store app on the Surface Hub to browse and download apps.

> **NOTE**
>
> Using the Microsoft Store app is not the recommended method of deploying apps at scale to your organization:
>
> - To download apps, you must sign in to the Microsoft Store app with a Microsoft account or organizational account. However, you can only connect an account to a maximum of 10 devices at once. If you have more than 10 Surface Hubs, you will need to create multiple accounts or remove devices from your account between app installations.
> - To install apps, you will need to manually sign in to the Microsoft Store app on each Surface Hub you own.

To browse the Microsoft Store on Surface Hub

1. From your Surface Hub, start **Settings**.
2. Type the device admin credentials when prompted.
3. Navigate to **This device** > **Apps & features**.
4. Select **Open Store**.

Download app packages from Microsoft Store for Business

To download the app package you need to install apps on your Surface Hub, visit the Microsoft Store for Business. The Store for Business is where you can find, acquire, and manage apps for the Windows 10 devices in your organization, including Surface Hub.

> **NOTE**
>
> Currently, Surface Hub only supports offline-licensed apps available through the Store for Business. App developers set offline-license availability when they submit apps.

Find and acquire the app you want, then download:

- The offline-licensed app package (either an .appx or an .appxbundle)
- The *unencoded* license file (if you're using provisioning packages to install the app)
- The *encoded* license file (if you're using MDM to distribute the app)
- Any necessary dependency files

For more information, see Download an offline-licensed app.

Provisioning package

You can manually install the offline-licensed apps that you downloaded from the Store for Business on a few Surface Hubs using provisioning packages. Use Windows Imaging and Configuration Designer (ICD) to create a provisioning package containing the app package and *unencoded* license file that you downloaded from the Store for Business. For more information, see Create provisioning packages.

Supported MDM provider

To deploy apps to a large number of Surface Hubs in your organization, use a supported MDM provider. The table below shows which MDM providers support deploying offline-licensed app packages.

MDM PROVIDER	SUPPORTS OFFLINE-LICENSED APP PACKAGES
On-premises MDM with System Center Configuration Manager (beginning in version 1602)	Yes
Hybrid MDM with System Center Configuration Manager and Microsoft Intune	Yes
Microsoft Intune standalone	No
Third-party MDM provider	Check to make sure your MDM provider supports deploying offline-licensed app packages.

To deploy apps remotely using System Center Configuration Manager (either on-prem MDM or hybrid MDM)

> **NOTE**
> These instructions are based on the current branch of System Center Configuration Manager.

1. Enroll your Surface Hubs to System Center Configuration Manager. For more information, see Enroll a Surface Hub into MDM.
2. Download the offline-licensed app package, the *encoded* license file, and any necessary dependency files from the Store for Business. For more information, see Download an offline-licensed app. Place the downloaded files in the same folder on a network share.
3. In the **Software Library** workspace of the Configuration Manager console, click **Overview** > **Application Management** > **Applications**.
4. On the **Home** tab, in the **Create** group, click **Create Application**.
5. On the **General** page of the **Create Application Wizard**, select the **Automatically detect information about this application from installation files** check box.
6. In the **Type** drop-down list, select **Windows app package (*.appx, *.appxbundle)**.
7. In the **Location** field, specify the UNC path in the form \server\share\filename for the offline-licensed app package that you downloaded from the Store for Business. Alternatively, click **Browse** to browse to the app package.

8. On the **Import Information** page, review the information that was imported, and then click **Next**. If necessary, you can click **Previous** to go back and correct any errors.
9. On the **General Information** page, complete additional details about the app. Some of this information might already be populated if it was automatically obtained from the app package.
10. Click **Next**, review the application information on the Summary page, and then complete the Create Application Wizard.
11. Create a deployment type for the application. For more information, see Create deployment types for the application.
12. Deploy the application to your Surface Hubs. For more information, see Deploy applications with System Center Configuration Manager.
13. As needed, update the app by downloading a new package from the Store for Business, and publishing an application revision in Configuration Manager. For more information, see Update and retire applications with System Center Configuration Manager.

> **NOTE**
> If you are using System Center Configuration Manager (current branch), you can bypass the above steps by connecting the Store for Business to System Center Configuration Manager. By doing so, you can synchronize the list of apps you've purchased with System Center Configuration Manager, view these in the Configuration Manager console, and deploy them like you would any other app. For more information, see Manage apps from the Microsoft Store for Business with System Center Configuration Manager.

Summary

There are a few different ways to install apps on your Surface Hub depending on whether you are developing apps, evaluating apps on a small number of devices, or deploying apps broadly to your oganization. This table summarizes the supported methods:

INSTALL METHOD	DEVELOPING APPS	EVALUATING APPS ON A FEW DEVICES	DEPLOYING APPS BROADLY TO YOUR ORGANIZATION
Visual Studio	X		
Provisioning package	X	X	
Microsoft Store app		X	
Supported MDM provider			X

Related topics

Manage Microsoft Surface Hub

Microsoft Surface Hub administrator's guide

End a Surface Hub meeting with I'm Done

5/4/2017 • 3 min to read • Edit Online

Surface Hub is a collaboration device designed to be used in meeting spaces by different groups of people. At the end of a meeting, users can tap **I'm Done** to clean up any sensitive data and prepare the device for the next meeting. Surface Hub will clean up, or reset, the following states:

- Applications
- Operating system
- User interface

This topic explains what **I'm Done** resets for each of these states.

Applications

When you start apps on Surface Hub, they are stored in memory and data is stored at the application level. Data is available to all users during that session (or meeting) until date is removed or overwritten. When **I'm done** is selected, Surface Hub application state is cleared out by closing applications, deleting browser history, resetting applications, and removing Skype logs.

Close applications

Surface Hub closes all visible windows, including Win32 and Universal Windows Platform (UWP) applications. The application close stage uses the multitasking view to query the visible windows. Win32 windows that do not close within a certain timeframe are closed using **TerminateProcess**.

Delete browser history

Surface Hub uses Delete Browser History (DBH) in Edge to clear Edge history and cached data. This is similar to how a user can clear out their browser history manually, but **I'm Done** also ensures that application states are cleared and data is removed before the next session, or meeting, starts.

Reset applications

I'm Done resets the state of each application that is installed on the Surface Hub. Resetting an application clears all background tasks, application data, notifications, and user consent dialogs. Applications are returned to their first-run state for the next people that use Surface Hub.

Remove Skype logs

Skype does not store personally-identifiable information on Surface Hub. Information is stored in the Skype service to meet existing Skype for Business guidance. Local Skype logging information is the only data removed when **I'm Done** is selected. This includes Unified Communications Client Platform (UCCP) logs and media logs.

Operating System

The operating system hosts a variety of information about the state of the sessions that needs to be cleared after each Surface Hub meeting.

File System

Meeting attendees have access to a limited set of directories on the Surface Hub. When **I'm Done** is selected, Surface Hub clears these directories:

- Music
- Videos

- Documents
- Pictures
- Downloads

Surface Hub also clears these directories, since many applications often write to them:

- Desktop
- Favorites
- Recent
- Public Documents
- Public Music
- Public Videos
- Public Downloads

Credentials

User credentials that are stored in **TokenBroker**, **PasswordVault**, or **Credential Manager** are cleared when you tap **I'm done**.

User interface

User interface (UI) settings are returned to their default values when **I'm Done** is selected.

UI items

- Reset Quick Actions to default state
- Clear Toast notifications
- Reset volume levels
- Reset sidebar width
- Reset tablet mode layout

Accessibility

Accessibility features and apps are returned to default settings when **I'm Done** is selected.

- Filter keys
- High contrast
- Sticky keys
- Toggle keys
- Mouse keys
- Magnifier
- Narrator

Clipboard

The clipboard is cleared to remove data that was copied to the clipboard during the session.

Frequently asked questions

What happens if I forget to tap I'm Done at the end of a meeting, and someone else uses the Surface Hub later?

Surface Hub only cleans up meeting content when users tap **I'm Done**. If you leave the meeting without tapping **I'm Done**, the device will return to the welcome screen after some time. From the welcome screen, users have the option to resume the previous session or start a new one.

Are documents recoverable?

Removing files from the hard drive when **I'm Done** is selected is just like any other file deletion from a hard disk

drive. Third-party software might be able to recover data from the hard disk drive, but file recovery is not a supported feature on Surface Hub. To prevent data loss, always save the data you need before leaving a meeting.

Do the clean-up actions from I'm Done comply with the US Department of Defense clearing and sanitizing standard: DoD 5220.22-M?

No. Currently, the clean-up actions from **I'm Done** do not comply with this standard.

Save your BitLocker key (Surface Hub)

5/4/2017 • 1 min to read • Edit Online

Every Microsoft Surface Hub is automatically set up with BitLocker drive encryption software. Microsoft strongly recommends that you make sure you back up your BitLocker recovery keys.

There are several ways to manage your BitLocker key on the Surface Hub.

1. If you've joined the Surface Hub to a domain, the device will back up the key on the domain and store it under the computer object.

   If you can't find the BitLocker key after joining the device to a domain, it's likely that your Active Directory schema doesn't support BitLocker key backup. If you don't want to change the schema, you can save the BitLocker key by going to Settings and following the procedure for using a local admin account, which is detailed later in this list.

2. If you've joined the Surface Hub to Azure Active Directory (Azure AD), the BitLocker key will be stored under the account that was used to join the device.

3. If you're using an admin account to manage the device, you can save the BitLocker key by going to the **Settings** app and navigating to **Update & security** > **Recovery**. Insert a USB drive and select the option to save the BitLocker key. The key will be saved to a text file on the USB drive.

Related topics

Manage Microsoft Surface Hub

Microsoft Surface Hub administrator's guide

Connect other devices and display with Surface Hub

5/4/2017 • 8 min to read • Edit Online

You can connect other devices to your Microsoft Surface Hub to display content. This topic describes the Guest Mode, Replacement PC Mode, and Video Out functionality available through wired connections, and also lists accessories that you can connect to Surface Hub using Bluetooth.

Which method should I choose?

When connecting external devices and displays to a Surface Hub, there are several available options. The method you use will depend upon your scenario and needs.

WHEN YOU WANT TO:	USE THIS METHOD:
Mirror the Surface Hub's display on another device.	Video Out
Present another device's display on the Surface Hub screen and interact with both the device's content and the built-in Surface Hub experience.	Guest Mode
Power the Surface Hub from an external Windows 10 PC, turning off the embedded computer of the Surface Hub. Cameras, microphones, speakers, and other peripherals, are sent to the external PC, in addition to pen and touch.	Replacement PC Mode

Guest Mode

Guest Mode uses a wired connection, so people can display content from their devices to the Surface Hub. If the source device is Windows-based, that device can also provide Touchback and Inkback. Surface Hub's internal PC takes video and audio from the connected device and presents them on the Surface Hub. If Surface Hub encounters a High-Bandwidth Digital Content Protection (HDCP) signal, the source will be re-routed through an alternate path, allowing the source to be displayed full-screen without violating HDCP requirements.

> **NOTE**
> When an HDCP source is connected, use the side keypad to change source inputs.

Ports

Use these ports on the Surface Hub for Guest Mode.

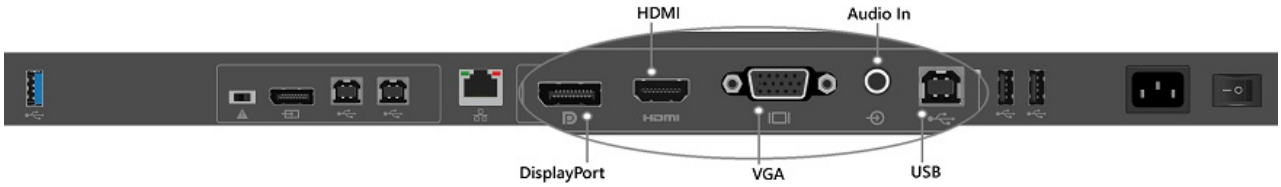
INTERFACE	TYPE	DESCRIPTION	CAPABILITIES

INTERFACE	TYPE	DESCRIPTION	CAPABILITIES
Display Port 1.1a	Video input	Guest input #1	• Support simultaneous guest input display with guest input #2 and guest input #3 (one full resolution, two thumbnail). • HDCP compliant in bypass mode • Touchback enabled
HDMI 1.4	Video input	Guest input #2	• Support simultaneous guest input display with guest input #1 and guest input #3 (one full resolution, two thumbnail). • HDCP compliant in bypass mode • Touchback enabled
VGA	Video input	Guest input #3	• Support simultaneous guest input display with guest input #1 and guest input #2 (one full resolution, two thumbnail). • HDCP compliant in bypass mode • Touchback enabled
3.5 mm jack	Audio input	Analog audio input	• Ingest into Surface Hub PC, usually with the VGA video input.

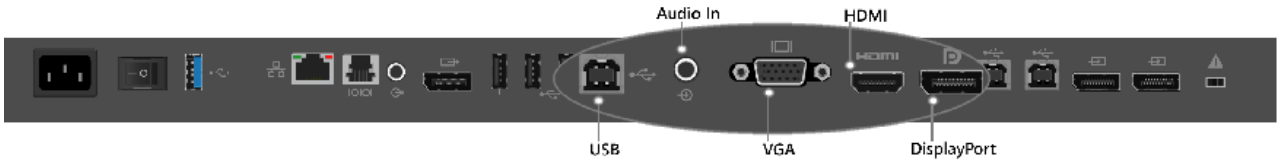
INTERFACE	TYPE	DESCRIPTION	CAPABILITIES
USB 2.0, type B	USB out	Touchback	• Provides access to the HID input devices mouse, touch, keyboard, and stylus back to the guest PC.

Port locations

These are the port connections used for Guest Mode on the 55" and 84" Surface Hubs.



Wired port connections on 55" Surface Hub



Wired port connections on 84" Surface Hub

Port enumeration

When a Surface hub is connected to a guest computer with the wired connect USB port, a number of USB devices are discovered and configured. These peripheral devices are created for Touchback and Inkback. The peripheral devices can be viewed in Device Manager. Device Manager will show duplicate names for some devices.

Human interface devices

- HID-compliant consumer control device
- HID-compliant pen
- HID-compliant pen (duplicate item)
- HID-compliant pen (duplicate item)
- HID-compliant touch screen
- USB Input Device
- USB Input Device (duplicate item)

Keyboards

- Standard PS/2 keyboard

Mice and other pointing devices

- HID-compliant mouse

Universal serial bus controllers

- Generic USB hub

- USB composite device

Guest Mode connectivity

Your choice of video cable will be determined by what is available from your source input. The Surface Hub has three choices of video input: DisplayPort, HDMI, and VGA. See the following chart for available resolutions.

SIGNAL TYPE	RESOLUTION	FRAME RATE	HDMI - RGB	DISPLAYPORT	VGA
PC	640 x 480	59.94/60	X	X	X
PC	720 x 480	59.94/60	X	X	
PC	1024 x 768	60	X	X	X
HDTV	720p	59.94/60	X	X	X
HDTV	1080p	59.94/60	X	X	X

Source audio is provided by DisplayPort and HDMI cables. If you must use VGA, Surface Hub has an audio input port that uses a 3.5 mm plug. Surface Hub also uses a USB cable that provides Touchback and Inkback from the Surface Hub to compatible Windows 10 devices. The USB cable can be used with any video input that is already connected with a cable.

Someone using Guest Mode to connect a PC would use one of these options:

DisplayPort -- DisplayPort cable and USB 2.0 cable

HDMI -- HDMI cable and USB 2.0 cable

VGA -- VGA cable, 3.5 mm audio cable, and USB 2.0 cable

If the computer you are using for Guest Mode is not compatible with Touchback and Inkback, then you won't need the USB cable.

Replacement PC Mode

In Replacement PC Mode, the embedded computer of the Surface Hub is turned off and an external PC is connected to the Surface Hub. Connections to replacement PC ports give access to key peripherals on the Surface Hub, including the screen, pen, and touch features. This does mean that your Surface Hub won't have the benefit of the Windows Team experience, but you will have the flexibility offered by providing and managing your own Windows computer.

Software requirements

You can run Surface Hub in Replacement PC Mode with 64-bit versions of Windows 10 Home, Windows 10 Pro, and Windows 10 Enterprise. You can download the Surface Hub Replacement PC driver package from the Microsoft Download Center. We recommend that you install these drivers on any computer you plan to use as a replacement PC.

Hardware requirements

Surface Hub is compatible with a range of hardware. Choose the processor and memory confirmation for your replacement PC so that it supports the programs you'll be using. Your replacement PC hardware needs to support 64-bit versions of Windows 10.

Graphics adapter

In Replacement PC Mode, Surface Hub supports any graphics adapter that can produce a DisplayPort signal. You'll improve your experience with a graphics adapter that can match Surface Hub's resolution and refresh rate. For example, the best and recommended replacement PC experience on the Surface Hub is with a 120Hz video signal.

55" Surface Hubs - For best experience, use a graphics card capable of 1080p resolution at 120Hz.

84" Surface Hubs - For best experience, use a graphics card capable of outputting four DisplayPort 1.2 streams to produce 2160p at 120Hz (3840 x 2160 at 120Hz vertical refresh). We've verified that this works with the NVIDIA Quadro K2200, NVIDIA Quadro K4200, NVIDIA Quadro M6000, AMD FirePro W5100, AMD FirePro W7100, and AMD FirePro W9100. These are not the only graphics cards - others are available from other vendors.

Check directly with graphics card vendors for the latest drivers.

GRAPHICS VENDOR	DRIVER DOWNLOAD PAGE
NVIDIA	http://nvidia.com/Download/index.aspx
AMD	http://support.amd.com/en-us/download
Intel	https://downloadcenter.intel.com/

Ports

Replacement PC ports on 55" Surface Hub

DESCRIPTION	TYPE	INTERFACE	DETAILS
PC video	Video input	DP 1.2	• Full screen display of 1080p at 120 Hz, plus audio • HDCP compliant
Internal peripherals	USB output	USB 2.0 type B	• Touch • Pen • Speakers • Microphone • Cameras • NFC sensor • Ambient light sensor • Passive infrared sensor

DESCRIPTION	TYPE	INTERFACE	DETAILS
USB hub	USB output	USB 2.0 type B	• Underneath USB ports

Replacement PC ports on 84" Surface Hub

DESCRIPTION	TYPE	INTERFACE	DETAILS
PC video	Video input	DP 1.2 (2x)	• Full screen display of 2160p at 120 Hz, plus audio • HDCP compliant
Internal peripherals	USB output	USB 2.0 type B	• Touch • Pen • Speakers • Microphone • Cameras • NFC sensor • Ambient light sensor • Passive infrared sensor
USB hub	USB output	USB 2.0 type B	• Underneath USB ports

Replacement PC setup instructions

To use Replacement PC Mode

1. Download and install the Surface Hub Replacement PC driver package on the replacement PC.

   > **NOTE**
   > We recommend that you set sleep or hibernation on the replacement PC so the Surface Hub will turn off the display when it isn't being used.

2. Turn off the Surface Hub using the power switch next to the power cable.
3. Connect the cables from the Surface Hub's replacement PC ports to the replacement PC. These ports are usually covered by a removable plastic cover.

55" Surface Hub -- connect one DisplayPort cable, and two USB cables.

84" Surface Hub -- connect two DisplayPort cables, and two USB cables.

4. Toggle the Mode switch to **Replacement PC**. The Mode switch is next to the Replacement PC ports.
5. Turn on the Surface Hub using the power switch next to the power cable.
6. Press the power button on the right side of the Surface Hub.

You can switch the Surface Hub to use the internal PC.

To switch back to internal PC

1. Turn off the Surface Hub using the power switch next to the power cable.
2. Toggle the Mode switch to Internal PC. The Mode switch is next to the Replacement PC ports.
3. Turn on the Surface Hub using the power switch next to the power cable.

Video Out

The Surface Hub includes a Video Out port for mirroring visual content from the Surface Hub to another display.

Ports

Video Out port on the 55" Surface Hub

Video Out port on the 84" Surface Hub

DESCRIPTION	TYPE	INTERFACE	CAPABILITIES

DESCRIPTION	TYPE	INTERFACE	CAPABILITIES
Video Output Mirror	Video Output	Video Output	• Supports connection to a standard DisplayPort monitor (only supports an x4 Link displaying 1080p60 resolution at 24bpp) • Supports use with HDMI monitors (supporting 1080p60) by using a DisplayPort-to-HDMI adaptor

Cables

Both the 55″ and 84″ Surface Hub devices have been tested to work with Certified DisplayPort and HDMI cables. While vendors do sell longer cables that may work with the Surface Hub, only those cables that have been certified by testing labs are certain to work with the Hub. For example, DisplayPort cables are certified only up to 3 meters, however many vendors sell cables that are 3 times that length. If a long cable is necessary, we strongly suggest using HDMI. HDMI has many cost-effective solutions for long-haul cables, including the use of repeaters. Nearly every DisplayPort source will automatically switch to HDMI signaling if a HDMI sink is detected.

Bluetooth accessories

You can connect the following accessories to Surface Hub using Bluetooth:

- Mice
- Keyboards
- Headsets
- Speakers

> **NOTE**
> After you connect a Bluetooth headset or speaker, you might need to change the default microphone and speaker settings.

Using a room control system (Surface Hub)

5/4/2017 • 4 min to read • Edit Online

Room control systems can be used with your Microsoft Surface Hub.

Using a room control system with your Surface Hub involves connecting room control hardware to the Surface Hub, usually through the RJ11 serial port on the bottom of the Surface Hub.

Terminal settings

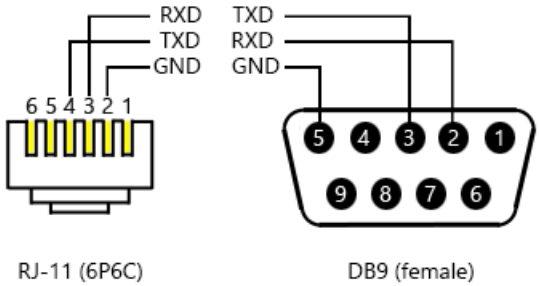
To connect to a room control system control panel, you don't need to configure any terminal settings on the Surface Hub. If you want to connect a PC or laptop to your Surface Hub and send serial commands from the Surface Hub, you can use a terminal emulator program like Tera Term or PuTTY.

SETTING	VALUE
Baud rate	115200
Data bits	8
Stop bits	1
Parity	none
Flow control	none
Line feed	every carriage return

Wiring diagram

You can use a standard RJ-11 (6P6C) connector to connect the Surface Hub serial port to a room control system. This is the recommended method. You can also use an RJ-11 4-conductor cable, but we do not recommend this method.

This diagram shows the correct pinout used for an RJ-11 (6P6C) to DB9 cable.



Command sets

Room control systems use common meeting-room scenarios for commands. Commands originate from the room control system, and are communicated over a serial connection to a Surface Hub. Commands are ASCII based, and the Surface Hub will acknowledge when state changes occur.

The following command modifiers are available. Commands terminate with a new line character (/n). Responses can come at any time in response to state changes not triggered directly by a management port command.

MODIFIER	RESULT
+	Increment a value
-	Decrease a value
=	Set a discrete value
?	Queries for a current value

Power

Surface Hub can be in one of these power states.

STATE	ENERGY STAR STATE	DESCRIPTION
0	S5	Off
1	-	Power up (indeterminate)
2	S3	Sleep
5	S0	Ready

In Replacement PC mode, the power states are only Ready and Off and only change the display. The management port can't be used to power on the replacement PC.

STATE	ENERGY STAR STATE	DESCRIPTION
0	S5	Off
5	S0	Ready

For a control device, anything other than 5 / Ready should be considered off. Each PowerOn command results in two state changes and reponses.

COMMAND	STATE CHANGE	RESPONSE
PowerOn	Device turns on (display + PC). PC service notifies SMC that the PC is ready.	Power=0 Power=5
PowerOff	Device transitions to ambient state (PC on, display dim).	Power=0
Power?	SMC reports the last-known power state.	Power=<#>

Brightness

The current brightness level is a range from 0 to 100.

Changes to brightness levels can be sent by a room control system, or other system.

COMMAND	STATE CHANGE	RESPONSE
Brightness+	System management controller (SMC) sends the brightness up command. PC service on the room control system notifies SMC of new brightness level.	Brightness = 51
Brightness-	SMC sends the brightness down command. PC service notifies SMC of new brightness level.	Brightness = 50

Volume

The current volume level is a range from 0 to 100.

Changes to volume levels can be sent by a room control system, or other system.

> **NOTE**
> The Volume command will only control the volume for embedded or Replacement PC mode, not from Guest sources.

COMMAND	STATE CHANGE	RESPONSE (ON IN REPLACEMENT PC MODE)
Volume+	SMC sends the volume up command. PC service notifies SMC of new volume level.	Volume = 51
Volume-	SMC sends the volume down command. PC service notifies SMC of new volume level.	Volume = 50

Mute for audio

Audio can be muted.

COMMAND	STATE CHANGE	RESPONSE
AudioMute+	SMC sends the audio mute command. PC service notifies SMC that audio is muted.	none

Video source

Several display sources can be used.

STATE	DESCRIPTION
0	Onboard PC

STATE	DESCRIPTION
1	DisplayPort
2	HDMI
3	VGA

Changes to display source can be sent by a room control system, or other system.

COMMAND	STATE CHANGE	RESPONSE
Source=#	SMC changes to the desired source. PC service notifies SMC that the display source has switched.	Source=<#>
Source+	SMC cycles to the next active input source. PC service notifies SMC of the current input source.	Source=<#>
Source-	SMC cycles to the previous active input source. PC service notifies SMC of the current input source.	Source=<#>
Source?	SMC queries PC service for the active input source. PC service notifies SMC of the current in;put source.	Source=<#>

Errors

Errors are returned following the format in this table.

ERROR	NOTES
Error: Unknown command '<input>'.	The instruction contains an unknown initial command. For example, "VOL+" would be invalid and return " Error: Unknown command 'VOL'".
Error: Unknown operator '<input>'.	The instruction contains an unknown operator. For example, "Volume!" would be invalid and return " Error: Unknown operator '!'".
Error: Unknown parameter '<input>'.	The instruction contains an unknown parameter. For example, "Volume=abc" would be invalid and return " Error: Unknown parameter 'abc'".
Error: Command not available when off '<input>'.	When the Surface Hub is off, commands other than Power return this error. For example, "Volume+" would be invalid and return " Error: Command not available when off 'Volume'".

Related topics

Manage Microsoft Surface Hub

Troubleshoot Microsoft Surface Hub

5/4/2017 • 9 min to read • Edit Online

Troubleshoot common problems, including setup issues, Exchange ActiveSync errors.

Common issues are listed in the following table, along with causes and possible fixes. The Setup troubleshooting section contains a listing of on-device problems, along with several types of issues that may be encountered during the first-run experience. The Exchange ActiveSync errors section lists common errors the device may encounter when trying to synchronize with an Microsoft Exchange ActiveSync server.

Setup troubleshooting

This section lists causes, and possible fixes to help troubleshoot issues you might find when you set up your Microsoft Surface Hub.

On-device

Possible fixes for issues on the Surface Hub after you've completed the first-run program.

ISSUE	CAUSES	POSSIBLE FIXES
Not receiving automatic accept/decline messages.	The device account isn't configured to automatically accept/decline messages.	Use PowerShell cmdlet `Set-CalendarProcessing $upn -AutomateProcessing AutoAccept` .
	The device account isn't configured to process external meeting requests.	Use PowerShell cmdlet `Set-CalendarProcessing $upn -ProcessExternalMeetingMessages $true` .
Calendar is not showing on the Welcome screen, or message "Appointments of date (no account provisioned)" is being displayed.	No device account is set up on this Surface Hub.	Provision a device account through Settings.
Calendar is not showing on the Welcome screen or message "Appointments of date (overprovisioned)" is being displayed.	The device account is provisioned on too many devices.	Remove the device account from other devices that it's provisioned to. This can be done using the Exchange admin portal.
Calendar is not showing on the Welcome screen or message "Appointments of date (invalid credentials)" is being displayed.	The device account's password has expired and is no longer valid.	Update the account's password in Settings. Also see Password management.

Calendar is not showing on the Welcome screen or message "Appointments of date (account policy)" is being displayed.	The device account is using an invalid ActiveSync policy.	Make sure the device account has an ActiveSync policy where `PasswordEnabled == False`.
Calendar is not showing on the Welcome screen or message "Appointments may be out of date" is being displayed.	Exchange is not enabled.	Enable the device account for Exchange services through Settings. You need to make sure you have the right set of ActiveSync policies and have also installed any necessary certificates for Exchange services to work.
Can't log in to Skype for Business.	The device account does not have a Session Initiation Protocol (SIP) address property.	The account does not have a SIP address property and its User Principal Name (UPN) does not match the actual SIP address. The account must have its SIP address set, or the SIP address should be added using the Settings app.
Can't log in to Skype for Business.	The device account requires a certificate to authenticate into Skype for Business.	Install the appropriate certificate using provisioning packages.

First run

Possible fixes for issues with Surface Hub first-run program.

ISSUE	CAUSES	POSSIBLE FIXES
Cannot find account when asked for domain and user name.	Domain needs to be the fully qualified domain name (FQDN).	The FQDN should be provided in the domain field.

Device account page, issues for new account settings

ISSUE	CAUSES	POSSIBLE FIXES
Unable to find the provided account in Azure AD.	The provided account's User Principal Name (UPN) has a tenant that can't be reached in Azure AD.	Make sure that you have a working Internet connection, and that the device can reach Microsoft Online Services. Make sure the account credentials are entered correctly.
Unable to reach the specified directory.	The provided account domain specifies a domain that can't be reached.	Make sure that you have a working network connection, and that the device can reach the domain controller. Make sure the account credentials are entered correctly. You can also try using the FQDN instead.

ISSUE	CAUSES	POSSIBLE FIXES
Can't auto-discover Exchange server.	The Exchange server isn't configured for auto-discovery.	Enable auto-discovery of the Exchange server for the device account, or enter the account's Exchange server address manually.
Could not discover the SIP address after entering the account credentials.	There was no SIP address entry in Active Directory or Azure AD.	Make sure the account is enabled with Skype for Business and has a SIP address. If not, you can enter the SIP address manually into the text box.

Device account page, issues for existing account settings

ISSUE	CAUSES	ERROR CODES	POSSIBLE FIXES
Account could not authenticate with the specified credentials.	The account is not enabled as a user in Active Directory (AD), needs a password to authenticate, or the password is incorrect.	None	Make sure the credentials are entered correctly. Enable the account as a user in AD and add a password, or set the RoomMailboxPassword .
Error 0x800C0019 is displayed when providing an Exchange server.	The device account requires a certificate to authenticate.	0x800C0019	Install the appropriate certificate using provisioning packages.
Device account credentials are not valid for the provided Exchange server.	The provided Exchange server is not where the device account's mailbox is hosted.	None	Make sure you are providing the correct Exchange mail server for the device account.
HTTP timeout while trying to reach Exchange server.		0x80072EE2	
Couldn't find the provided Exchange server.	The Exchange server provided could not be found.	None	Ensure that you have a working network or Internet connection, and that the Exchange server you provided is correct.
http not supported.	An Exchange server with *http://* instead of *https://* was provided.	None	Use an Exchange server that uses https.

People land on the page titled "There's a problem with this account" regarding ActiveSync.	The ActiveSync policy PasswordEnabled is set to True (or 1).	None	Create a new ActiveSync policy where PasswordEnabled is set to False (or 0), and then apply that policy to the account.
	The Surface Hub doesn't have a connection to Exchange.	None	Make sure that you have a working network or Internet connection.
	Exchange returns a status code indicating an error.	None	Make sure that you have a working network or Internet connection.

First run, Domain join page issues

ISSUE	CAUSES	POSSIBLE FIXES
When trying to join a domain, an error shows that the account couldn't authenticate using the specified credentials.	The credentials provided are not capable of joining the specified domain.	Enter correct credentials for an account that exists in the specified domain.
When specifying a group from a domain, an error shows that the group couldn't be found on the domain.	The group may have been removed or no longer exists.	Verify that the group exists within the domain.

First run, Exchange server page

ISSUE	CAUSES	POSSIBLE FIXES
People land on this page and are asked for the Exchange server address.	The Exchange server isn't configured for auto-discovery.	Enable auto-discovery of the Exchange server for the device account, or enter the account's Exchange server address manually.

First run, On-device issues

ISSUE	CAUSES	ERROR CODES	POSSIBLE FIXES
Can't sync mail/calendar.	The account has not allowed the Surface Hub as an allowed device.	0x86000C1C	Add the Surface Hub device ID to the whitelist by setting the **ActiveSyncAllowedDeviceIds** property for the mailbox.

Skype for Business

ISSUE	CAUSES	POSSIBLE FIXES
Can't call a Skype consumer from my Surface Hub.	Outgoing calls aren't supported yet.	None currently.

Exchange ActiveSync errors

This section lists status codes, mapping, user messages, and actions an admin can take to solve Exchange ActiveSync errors.

HEX CODE	MAPPING	USER-FRIENDLY MESSAGE	ACTION ADMIN SHOULD TAKE
0x85010002	E_HTTP_DENIED	The password must be updated.	Update the password.
0x80072EFD	WININET_E_CANNOT_CONNECT	Can’t connect to the server right now. Wait a while and try again, or check the account settings.	Verify that the server name is correct and reachable. Verify that the device is connected to the network.
0x86000C29	E_NEXUS_STATUS_DEVICE_NOTPROVISIONED (policies don’t match)	The account is configured with policies not compatible with Surface Hub.	Disable the **PasswordEnabled** policy for this account. We have a bug were we may surface policy errors if the account doesn’t receive any server notifications within the policy refresh interval.
0x86000C4C	E_NEXUS_STATUS_MAXIMUMDEVICESREACHED	The account has too many device partnerships.	Delete one or more partnerships on the server.
0x86000C0A	E_NEXUS_STATUS_SERVERERROR_RETRYLATER	Can’t connect to the server right now.	Wait until the server comes back online. If the issue persists, re-provision the account.
0x85050003	E_CREDENTIALS_EXPIRED (Credentials have expired and need to be updated)	The password must be updated.	Update the password.

HEX CODE	MAPPING	USER-FRIENDLY MESSAGE	ACTION ADMIN SHOULD TAKE
0x8505000D	E_AIRSYNC_RESET_RETRY	Can't connect to the server right now. Wait a while or check the account's settings.	This is normally a transient error but if the issue persists check the number of devices associated with the account and delete some of them if the number is large.
0x86000C16	E_NEXUS_STATUS_USER_HASNOMAILBOX	The mailbox was migrated to a different server.	You should never see this error. If the issue persists, re-provision the account.
0x85010004	E_HTTP_FORBIDDEN	Can't connect to the server right now. Wait a while and try again, or check the account's settings.	Verify the server name to make sure it is correct. If the account is using cert based authentication make sure the certificate is still valid and update it if not.
0x85030028	E_ACTIVESYNC_PASSWORD_OR_GETCERT	The account's password or client certificate are missing or invalid.	Update the password and/or deploy the client certificate.
0x86000C2A	E_NEXUS_STATUS_DEVICE_POLICYREFRESH	The account is configured with policies not compatible with Surface Hub.	Disable the PasswordEnabled policy for this account.
0x85050002	E_CREDENTIALS_UNAVAILABLE	The password must be updated.	Update the password.
0x80072EE2	WININET_E_TIMEOUT	The network doesn't support the minimum idle timeout required to receive server notification, or the server is offline.	Verify that the server is running. Verify the NAT settings.
0x85002004	E_FAIL_ABORT	This error is used to interrupt the hanging sync, and will not be exposed to users. It will be shown in the telemetry if you force an interactive sync, delete the account, or update its settings.	Nothing.

HEX CODE	MAPPING	USER-FRIENDLY MESSAGE	ACTION ADMIN SHOULD TAKE
0x85010017	E_HTTP_SERVICE_UNAVAIL	Can't connect to the server right now. Wait a while or check the account's settings.	Verify the server name to make sure it is correct. Wait until the server comes back online. If the issue persists, re-provision the account.
0x86000C0D	E_NEXUS_STATUS_MAILBOX_SERVEROFFLINE	Can't connect to the server right now. Wait a while or check the account's settings.	Verify the server name to make sure it is correct. Wait until the server comes back online. If the issue persists, re-provision the account.
0x85030027	E_ACTIVESYNC_GETCERT	The Exchange server requires a certificate.	Import the appropriate EAS certificate on the Surface Hub.
0x86000C2B	E_NEXUS_STATUS_INVALID_POLICYKEY	The account is configured with policies not compatible with Surface Hub.	Disable the PasswordEnabled policy for this account. We have a bug were we may surface policy errors if the account doesn't receive any server notifications within the policy refresh interval.
0x85010005	E_HTTP_NOT_FOUND	The server name is invalid.	Verify the server name to make sure it is correct. If the issue persists, re-provision the account.
0x85010014	E_HTTP_SERVER_ERROR	Can't connect to the server.	Verify the server name to make sure it is correct. Trigger a sync and, if the issue persists, re-provision the account.
0x80072EE7	WININET_E_NAME_NOT_RESOLVED	The server name or address could not be resolved.	Make sure the server name is entered correctly.

HEX CODE	MAPPING	USER-FRIENDLY MESSAGE	ACTION ADMIN SHOULD TAKE
0x8007052F	ERROR_ACCOUNT_RESTRICTION	While auto-discovering the Exchange server, a policy is applied that prevents the logged-in user from logging in to the server.	This is a timing issue. Re-verify the account's credentials. Try to re-provision when they're correct.
0x800C0019	INET_E_INVALID_CERTIFICATE	Security certificate required to access this resource is invalid.	Install the correct ActiveSync certificate needed for the provided device account.
0x80072F0D	WININET_E_INVALID_CA	The certificate authority is invalid or is incorrect. Could not auto-discover the Exchange server because a certificate is missing.	Install the correct ActiveSync certificate needed for the provided device account.
0x80004005	E_FAIL	The domain provided couldn't be found. The Exchange server could not be auto-discovered and was not provided in the settings.	Make sure that the domain entered is the FQDN, and that there is an Exchange server entered in the Exchange server text box.

Appendix: PowerShell (Surface Hub)

5/4/2017 • 31 min to read • Edit Online

PowerShell scripts to help set up and manage your Microsoft Surface Hub .

You can check online for updated versions at Surface Hub device account scripts.

PowerShell scripts for Surface Hub administrators

What do the scripts do?

- Create device accounts for setups using pure single-forest on-premises (Microsoft Exchange and Skype 2013 and later only) or online (Microsoft Office 365), that are configured correctly for your Surface Hub.
- Validate existing device accounts for any setup (on-premises or online) to make sure they're compatible with Surface Hub.
- Provide a base template for anyone wanting to create their own device account creation or validation scripts.

What do you need in order to run the scripts?

- Remote PowerShell access to your organization's domain or tenant, Exchange servers, and Skype for Business servers.
- Admin credentials for your organization's domain or tenant, Exchange servers, and Skype for Business servers.

> **Note** Whether you're creating a new account or modifying an already-existing account, the validation script will verify that your device account is configured correctly. You should always run the validation script before adding a device account to Surface Hub.

Running the scripts

The account creation scripts will:

- Ask for administrator credentials
- Create device accounts in your domain/tenant
- Create or assign a Surface Hub-compatible ActiveSync policy to the device account(s)
- Set various attributes for the created account(s) in Exchange and Skype for Business.
- Assign licenses and permissions to the created account(s)

These are the attributes that are set by the scripts:

CMDLET	ATTRIBUTE	VALUE
Set-Mailbox	RoomMailboxPassword	User-provided
	EnableRoomMailboxAccount	True
	Type	Room
Set-CalendarProcessing	AutomateProcessing	AutoAccept
	RemovePrivateProperty	False
	DeleteSubject	False
	DeleteComments	False
	AddOrganizerToSubject	False
	AddAdditionalResponse	True
	AdditionalResponse	"This is a Surface Hub room!"
New-MobileDeviceMailboxPolicy	PasswordEnabled	False
	AllowNonProvisionableDevices	True
Enable-CSMeetingRoom	RegistrarPool	User-provided
	SipAddress	Set to the User Principal Name (UPN) of the device account
Set-MsolUserLicense (O365 only)	AddLicenses	User-provided
Set-MsolUser (O365 only)	PasswordNeverExpires	True
Set-AdUser (On-prem only)	Enabled	True
Set-AdUser (On-prem only)	PasswordNeverExpires	True

Account creation scripts

These scripts will create a device account for you. You can use the Account verification script to make sure they ran correctly.

The account creation scripts cannot modify an already existing account, but can be used to help you understand which cmdlets need to be run to configure the existing account correctly.

Create an on-premise account

Creates an account as described in On-premises deployment.

```
# SHAccountCreateOnPrem.ps1

$Error.Clear()
$ErrorActionPreference = "Stop"
$status = @{}

# Cleans up set state such as remote powershell sessions
function Cleanup()
{
    if ($sessExchange)
    {
        Remove-PSSession $sessExchange
    }
    if ($sessCS)
    {
        Remove-PSSession $sessCS
    }
}

function PrintError($strMsg)
{
    Write-Host $strMsg -foregroundcolor Red
}

function PrintSuccess($strMsg)
{
    Write-Host $strMsg -foregroundcolor Green
}

function PrintAction($strMsg)
{
    Write-Host $strMsg -ForegroundColor Cyan
}


# Cleans up and prints an error message
function CleanupAndFail($strMsg)
{
    if ($strMsg)
    {
        PrintError($strMsg);
    }
    Cleanup
    exit 1
}

# Exits if there is an error set and prints the given message
function ExitIfError($strMsg)
{
    if ($Error)
    {
        CleanupAndFail($strMsg);
    }
}

## Collect account data ##
$credNewAccount = (Get-Credential -Message "Enter the desired UPN and password for this new account")
$strUpn = $credNewAccount.UserName
```

```
$strUpn = $credNewAccount.UserName
$strDisplayName = Read-Host "Please enter the display name you would like to use for $strUpn"
if (!$credNewAccount -Or [System.String]::IsNullOrEmpty($strDisplayName) -Or
[System.String]::IsNullOrEmpty($credNewAccount.UserName) -Or $credNewAccount.Password.Length -le 0)
{
    CleanupAndFail "Please enter all of the requested data to continue."
    exit 1
}


## Sign in to remote powershell for exchange and lync online ##

$credExchange = $null
$credExchange=Get-Credential -Message "Enter credentials of an Exchange user with mailbox creation rights"
if (!$credExchange)
{
    CleanupAndFail("Valid credentials are required to create and prepare the account.");
}
$strExchangeServer = Read-Host "Please enter the FQDN of your exchange server (e.g. exch.contoso.com)"

# Lync info
$credLync = Get-Credential -Message "Enter credentials of a Skype for Business admin (or cancel if they are
the same as Exchange)"
if (!$credLync)
{
    $credLync = $credExchange
}
$strLyncFQDN = Read-Host "Please enter the FQDN of your Lync server (e.g. lync.contoso.com) or enter to use
[$strExchangeServer]"
if ([System.String]::IsNullOrEmpty($strLyncFQDN))
{
    $strLyncFQDN = $strExchangeServer
}


PrintAction "Connecting to remote sessions. This can occasionally take a while - please do not enter input..."
try
{
    $sessExchange = New-PSSession -ConfigurationName microsoft.exchange -Credential $credExchange -
AllowRedirection -Authentication Kerberos -ConnectionUri "http://$strExchangeServer/powershell" -WarningAction
SilentlyContinue
}
catch
{
    CleanupAndFail("Failed to connect to exchange. Please check your credentials and try again. If this
continues to fail, you may not have permission for remote powershell - if not, please perform the setup
manually. Error message: $_")
}
PrintSuccess "Connected to Remote Exchange Shell"

try
{
    $sessLync = New-PSSession -Credential $credLync -ConnectionURI "https://$strLyncFQDN/OcsPowershell" -
AllowRedirection -WarningAction SilentlyContinue
}
catch
{
    CleanupAndFail("Failed to connect to Lync. Please check your credentials and try again. Error message:
$_")
}
PrintSuccess "Connected to Lync Server Remote PowerShell"


Import-PSSession $sessExchange -AllowClobber -WarningAction SilentlyContinue
Import-PSSession $sessLync -AllowClobber -WarningAction SilentlyContinue

# In case there was any uncaught errors
ExitIfError("Remote connections failed. Please check your credentials and try again.")
```

```
## Create the Exchange mailbox ##
# Note: These exchange commandlets do not always throw their errors as exceptions

# Because Get-Mailbox will throw an error if the mailbox is not found
$Error.Clear()
PrintAction "Creating a new account..."
try
{
    $mailbox = $null
    $mailbox = (New-Mailbox -UserPrincipalName $credNewAccount.UserName -Alias
$credNewAccount.UserName.substring(0,$credNewAccount.UserName.indexOf('@')) -room -Name $strDisplayName -
RoomMailboxPassword $credNewAccount.Password -EnableRoomMailboxAccount $true)
} catch { }
ExitIfError "Failed to create a new mailbox on exchange.";
$status["Mailbox Setup"] = "Successfully created a mailbox for the new account"


$strEmail = $mailbox.WindowsEmailAddress
PrintSuccess "The following mailbox has been created for this room: $strEmail"


## Create or retrieve a policy that will be applied to surface hub devices ##
# The policy disables requiring a device password so that the SurfaceHub does not need to be lockable to use
Active Sync
$strPolicy = Read-Host 'Please enter the name for a new Surface Hub ActiveSync policy that will be created and
applied to this account.
We will configure that policy to be compatible with Surface Hub devices.
If this script has been used before, please enter the name of the existing policy.'

$easpolicy = $null
try {
    $easpolicy = Get-MobileDeviceMailboxPolicy $strPolicy
}
catch {}

if ($easpolicy)
{
    if (!$easpolicy.PasswordEnabled -and ($easpolicy.AllowNonProvisionableDevices -eq $null -or
$easpolicy.AllowNonProvisionableDevices ))
    {
        PrintSuccess "An existing policy has been found and will be applied to this account."
    }
    else
    {
        PrintError "The policy you provided is incompatible with the surface hub."
        $easpolicy = $null
        $status["Device Password Policy"] = "Failed to apply the EAS policy to the account because the policy
was invalid."
    }
}
else
{
    $Error.Clear()
    PrintAction "Creating policy..."
    $easpolicy = New-MobileDeviceMailboxPolicy -Name $strPolicy -PasswordEnabled $false -
AllowNonProvisionableDevices $true
    if ($easpolicy)
    {
        PrintSuccess "A new device policy has been created; you can use this same policy for all future
Surface Hub device accounts."
    }
    else
    {
        PrintError "Could not create $strPolicy"
    }
}
```

```
if ($easpolicy)
{
    # Convert mailbox to user type so we can apply the policy (necessary)
    # Sometimes it takes a while for this change to take affect so we have some nasty retry loops
    $Error.Clear();
    try
    {
        Set-Mailbox $credNewAccount.UserName -Type Regular
    } catch {}
    if ($Error)
    {
        $Error.Clear()
        $status["Device Password Policy"] = "Failed to apply the EAS policy to the account."
    }
    else
    {
        # Loop until resource type goes away, up to 5 times
        for ($i = 0; $i -lt 5 -And (Get-Mailbox $credNewAccount.UserName).ResourceType; $i++)
        {
            Start-Sleep -s 5
        }
        # If the mailbox is still a Room we cannot apply the policy
        if (!((Get-Mailbox $credNewAccount.UserName).ResourceType))
        {
            $Error.Clear()
            # Set policy for account
            Set-CASMailbox $credNewAccount.UserName -ActiveSyncMailboxPolicy $strPolicy
            if (!$Error)
            {
                $status["ActiveSync Policy"] = "Successfully applied $strPolicy to the account"
            }
            else
            {
                $status["ActiveSync Policy"] = "Failed to apply the EAS policy to the account."
            }
            $Error.Clear()

            # Convert back to room mailbox
            Set-Mailbox $credNewAccount.UserName -Type Room
            # Loop until resource type goes back to room
            for ($i = 0; ($i -lt 5) -And ((Get-Mailbox $credNewAccount.UserName).ResourceType -ne "Room");
$i++)
            {
                Start-Sleep -s 5
            }
            if ((Get-Mailbox $credNewAccount.UserName).ResourceType -ne "Room")
            {
                # A failure to convert the mailbox back to a room is unfortunate but means the mailbox is
unusable.
                $status["Mailbox Setup"] = "A mailbox was created but we could not set it to a room resource
type."
            }
            else
            {
                try
                {
                    Set-Mailbox $credNewAccount.UserName -RoomMailboxPassword $credNewAccount.Password -
EnableRoomMailboxAccount $true
                } catch { }
                if ($Error)
                {
                    $status["Mailbox Setup"] = "A room mailbox was created but we could not set its password."
                }
                $Error.Clear()
            }

        }
    }
}
```

```
PrintSuccess "Account creation completed."

PrintAction "Setting calendar processing rules..."

$Error.Clear();
## Prepare the calendar for automatic meeting responses ##
try {
    Set-CalendarProcessing -Identity $credNewAccount.UserName -AutomateProcessing AutoAccept
} catch { }
if ($Error)
{
    $status["Calendar Acceptance"] = "Failed to configure the account to automatically accept/decline meeting
requests"
}
else
{
    $status["Calendar Acceptance"] = "Successfully configured the account to automatically accept/decline
meeting requests"
}


$Error.Clear()
try {
    Set-CalendarProcessing -Identity $credNewAccount.UserName -RemovePrivateProperty $false -
AddOrganizerToSubject $false -AddAdditionalResponse $true -DeleteSubject $false -DeleteComments $false -
AdditionalResponse "This is a Surface Hub room!"
} catch { }
if ($Error)
{
    $status["Calendar Response Configuration"] = "Failed to configure the account's response properties"
}
else
{
    $status["Calendar Response Configuration"] = "Successfully configured the account's response properties"
}

$Error.Clear()
## Configure the Account to not expire ##
PrintAction "Configuring password not to expire..."
Start-Sleep -s 20
try
{
    Set-AdUser $mailbox.Alias -PasswordNeverExpires $true -Enabled $true
}
catch
{

}

if ($Error)
{
    $status["Password Expiration Policy"] = "Failed to set the password to never expire"
}
else
{
    $status["Password Expiration Policy"] = "Successfully set the password to never expire"
}

PrintSuccess "Completed Exchange configuration"

## Setup Skype for Business. This is somewhat optional and if it fails we SfbEnable can be used later ##
PrintAction "Configuring account for Skype for Business."

# Getting registrar pool
$strRegPool = $strLyncFQDN
$Error.Clear()
$strRegPoolEntry = Read-Host "Enter a Skype for Business Registrar Pool, or leave blank to use [$strRegPool]"
if (![System.String]::IsNullOrEmpty($strRegPoolEntry))
{
```

```
    $strRegPool = $strRegPoolEntry
}

# Try to SfB-enable the account. Note that it may not work right away as the account needs to propogate to
active directory
PrintAction "Enabling Skype for Business..."
Start-Sleep -s 10
$Error.Clear()
try {
    Enable-CsMeetingRoom -Identity $credNewAccount.UserName -RegistrarPool $strRegPool -SipAddressType
EmailAddress
}
catch { }

if ($Error)
{
    $status["Skype for Business Account Setup"] = "Failed to setup the Skype for Business meeting room - you
can run EnableSfb.ps1 to try again."
    $Error.Clear();
}
else
{
    $status["Skype for Business Account Setup"] = "Successfully enabled account as a Skype for Business
meeting room"
}

Write-Host

## Cleanup and print results ##
Cleanup
$strDisplay = $mailbox.DisplayName
$strUsr = $credNewAccount.UserName
PrintAction "Summary for creation of $strUsr ($strDisplay)"
if ($status.Count -gt 0)
{
    ForEach($k in $status.Keys)
    {
        $v = $status[$k]
        $color = "yellow"
        if ($v[0] -eq "S") { $color = "green" }
        elseif ($v[0] -eq "F")
        {
            $color = "red"
            $v += " Go to http://aka.ms/shubtshoot"
        }

        Write-Host -NoNewline $k -ForegroundColor $color
        Write-Host -NoNewline ": "
        Write-Host $v
    }
}
else
{
    PrintError "The account could not be created"
}
```

Create a device account using Office 365

Creates an account as described in Create a device account using Office 365

```
# SHAccountCreateO365.ps1

$Error.Clear()
$ErrorActionPreference = "Stop"
$status = @{}

# Cleans up set state such as remote powershell sessions
```

```
function Cleanup()
{
    if ($sessExchange)
    {
        Remove-PSSession $sessExchange
    }
    if ($sessCS)
    {
        Remove-PSSession $sessCS
    }
}

function PrintError($strMsg)
{
    Write-Host $strMsg -foregroundcolor Red
}

function PrintSuccess($strMsg)
{
    Write-Host $strMsg -foregroundcolor Green
}

function PrintAction($strMsg)
{
    Write-Host $strMsg -ForegroundColor Cyan
}


# Cleans up and prints an error message
function CleanupAndFail($strMsg)
{
    if ($strMsg)
    {
        PrintError($strMsg);
    }
    Cleanup
    exit 1
}

# Exits if there is an error set and prints the given message
function ExitIfError($strMsg)
{
    if ($Error)
    {
        CleanupAndFail($strMsg);
    }
}


## Check dependencies ##
try {
    Import-Module LyncOnlineConnector
    Import-Module MSOnline
}
catch
{
    PrintError "Some dependencies are missing"
    PrintError "Please install the Windows PowerShell Module for Lync Online. For more information go to
http://www.microsoft.com/download/details.aspx?id=39366"
    PrintError "Please install the Azure Active Directory module for PowerShell from
https://go.microsoft.com/fwlink/p/?linkid=236297"
    CleanupAndFail
}



## Collect account data ##
$credNewAccount = (Get-Credential -Message "Enter the desired UPN and password for this new account")
$strUpn = $credNewAccount.UserName
```

```
$strDisplayName = Read-Host "Please enter the display name you would like to use for $strUpn"
if (!$credNewAccount -Or [System.String]::IsNullOrEmpty($strDisplayName) -Or
[System.String]::IsNullOrEmpty($credNewAccount.UserName) -Or $credNewAccount.Password.Length -le 0)
{
    CleanupAndFail "Please enter all of the requested data to continue."
    exit 1
}


## Sign in to remote powershell for exchange and lync online ##
$credAdmin = $null
$credAdmin=Get-Credential -Message "Enter credentials of an Exchange and Skype for Business admin"
if (!$credadmin)
{
    CleanupAndFail "Valid admin credentials are required to create and prepare the account."
}
PrintAction "Connecting to remote sessions. This can occasionally take a while - please do not enter input..."
try
{
    $sessExchange = New-PSSession -ConfigurationName microsoft.exchange -Credential $credAdmin -
AllowRedirection -Authentication basic -ConnectionUri "https://outlook.office365.com/powershell-liveid/" -
WarningAction SilentlyContinue
}
catch
{
    CleanupAndFail "Failed to connect to exchange. Please check your credentials and try again. Error message:
$_"
}

try
{
    $sessCS = New-CsOnlineSession -Credential $credAdmin
}
catch
{
    CleanupAndFail "Failed to connect to Skype for Business Online Datacenter. Please check your credentials
and try again. Error message: $_"
}

try
{
    Connect-MsolService -Credential $credAdmin
}
catch
{
    CleanupAndFail "Failed to connect to Azure Active Directory. Please check your credentials and try again.
Error message: $_"
}

Import-PSSession $sessExchange -AllowClobber -WarningAction SilentlyContinue
Import-PSSession $sessCS -AllowClobber -WarningAction SilentlyContinue

# In case there was any uncaught errors
ExitIfError "Remote connection failed. Please check your credentials and try again."



## Create the Exchange mailbox ##
# Note: These exchange commandlets do not always throw their errors as exceptions

# Because Get-Mailbox will throw an error if the mailbox is not found
$Error.Clear()
PrintAction "Creating a new account..."
try
{
    $mailbox = $null
    $mailbox = (New-Mailbox -MicrosoftOnlineServicesID $credNewAccount.UserName -room -Name $strDisplayName -
RoomMailboxPassword $credNewAccount.Password -EnableRoomMailboxAccount $true)
} catch { }
```

```
ExitIfError "Failed to create a new mailbox on exchange.";
$status["Mailbox Setup"] = "Successfully created a mailbox for the new account"


$strEmail = $mailbox.WindowsEmailAddress
PrintSuccess "The following mailbox has been created for this room: $strEmail"


## Create or retrieve a policy that will be applied to surface hub devices ##
# The policy disables requiring a device password so that the SurfaceHub does not need to be lockable to use
Active Sync
$strPolicy = Read-Host 'Please enter the name for a new Surface Hub ActiveSync policy that will be created and
applied to this account.
We will configure that policy to be compatible with Surface Hub devices.
If this script has been used before, please enter the name of the existing policy.'

$easpolicy = $null
try {
    $easpolicy = Get-MobileDeviceMailboxPolicy $strPolicy
}
catch {}

if ($easpolicy)
{
    if (!$easpolicy.PasswordEnabled -and ($easpolicy.AllowNonProvisionableDevices -eq $null -or
$easpolicy.AllowNonProvisionableDevices ))
    {
        PrintSuccess "An existing policy has been found and will be applied to this account."
    }
    else
    {
        PrintError "The policy you provided is incompatible with the surface hub."
        $easpolicy = $null
        $status["ActiveSync Policy"] = "Failed to apply the EAS policy to the account because the policy was
invalid."
    }
}
else
{
    $Error.Clear()
    PrintAction "Creating policy..."
    $easpolicy = New-MobileDeviceMailboxPolicy -Name $strPolicy -PasswordEnabled $false -
AllowNonProvisionableDevices $true
    if ($easpolicy)
    {
        PrintSuccess "A new device policy has been created; you can use this same policy for all future
Surface Hub device accounts."
    }
    else
    {
        PrintError "Could not create $strPolicy"
    }
}

if ($easpolicy)
{
    # Convert mailbox to user type so we can apply the policy (necessary)
    # Sometimes it takes a while for this change to take affect so we have some nasty retry loops
    $Error.Clear();
    try
    {
        Set-Mailbox $credNewAccount.UserName -Type Regular
    } catch {}
    if ($Error)
    {
        $Error.Clear()
        $status["Device Password Policy"] = "Failed to apply the EAS policy to the account."
        PrintError "Failed to convert to regular account"
    }
```

```
    }
    else
    {
        # Loop until resource type goes away, up to 5 times
        for ($i = 0; $i -lt 5 -And (Get-Mailbox $credNewAccount.UserName).ResourceType; $i++)
        {
            Start-Sleep -s 5
        }
        # If the mailbox is still a Room we cannot apply the policy
        if (!((Get-Mailbox $credNewAccount.UserName).ResourceType))
        {
            $Error.Clear()
            # Set policy for account
            Set-CASMailbox $credNewAccount.UserName -ActiveSyncMailboxPolicy $strPolicy
            if (!$Error)
            {
                $status["Device Password Policy"] = "Successfully applied $strPolicy to the account"
            }
            else
            {
                $status["Device Password Policy"] = "Failed to apply the EAS policy to the account."
                PrintError "Failed to apply policy"
            }
            $Error.Clear()

            # Convert back to room mailbox
            Set-Mailbox $credNewAccount.UserName -Type Room
            # Loop until resource type goes back to room
            for ($i = 0; ($i -lt 5) -And ((Get-Mailbox $credNewAccount.UserName).ResourceType -ne "Room");
$i++)
            {
                Start-Sleep -s 5
            }
            if ((Get-Mailbox $credNewAccount.UserName).ResourceType -ne "Room")
            {
                # A failure to convert the mailbox back to a room is unfortunate but means the mailbox is
unusable.
                $status["Mailbox Setup"] = "A mailbox was created but we could not set it to a room resource
type."
            }
            else
            {
                Set-Mailbox $credNewAccount.UserName -RoomMailboxPassword $credNewAccount.Password -
EnableRoomMailboxAccount $true
                if ($Error)
                {
                    $status["Mailbox Setup"] = "A room mailbox was created but we could not set its password."
                }
                $Error.Clear()
            }

        }
    }
}
else
{
    $status["Device Password Policy"] = "Failed to apply the EAS policy to the account."
    PrintError "Failed to obtain policy"
}
PrintSuccess "Account creation completed."

PrintAction "Setting calendar processing rules..."

$Error.Clear();
## Prepare the calendar for automatic meeting responses ##
try {
    Set-CalendarProcessing -Identity $credNewAccount.UserName -AutomateProcessing AutoAccept
} catch { }
if ($Error)
{
```

```
{
    $status["Calendar Acceptance"] = "Failed to configure the account to automatically accept/decline meeting
requests"
}
else
{
    $status["Calendar Acceptance"] = "Successfully configured the account to automatically accept/decline
meeting requests"
}


$Error.Clear()
try {
    Set-CalendarProcessing -Identity $credNewAccount.UserName -RemovePrivateProperty $false -
AddOrganizerToSubject $false -AddAdditionalResponse $true -DeleteSubject $false -DeleteComments $false -
AdditionalResponse "This is a Surface Hub room!"
} catch { }
if ($Error)
{
    $status["Calendar Response Configuration"] = "Failed to configure the account's response properties"
}
else
{
    $status["Calendar Response Configuration"] = "Successfully configured the account's response properties"
}

$Error.Clear()
## Configure the Account to not expire ##
PrintAction "Configuring password not to expire..."
try
{
    Set-MsolUser -UserPrincipalName $credNewAccount.UserName -PasswordNeverExpires $true
}
catch
{

}

if ($Error)
{
    $status["Password Expiration Policy"] = "Failed to set the password to never expire"
}
else
{
    $status["Password Expiration Policy"] = "Successfully set the password to never expire"
}

PrintSuccess "Completed Exchange configuration"

## Setup Skype for Business. This is somewhat optional and if it fails we SfbEnable can be used later ##
PrintAction "Configuring account for Skype for Business."

# Getting registrar pool
$strRegPool = $null
try {
    $strRegPool = (Get-CsTenant).TenantPoolExtension
}
catch {}
$Error.Clear()
if (![System.String]::IsNullOrEmpty($strRegPool))
{
    $strRegPool = $strRegPool.Substring($strRegPool[0].IndexOf(':') + 1)
}
<#
$strRegPoolEntry = Read-Host "Enter a Skype for Business Registrar Pool, or leave blank to use [$strRegPool]"
if (![System.String]::IsNullOrEmpty($strRegPoolEntry))
{
    $strRegPool = $strRegPoolEntry
}
```

```
#>

# Try to SfB-enable the account. Note that it may not work right away as the account needs to propogate to
active directory
PrintAction "Enabling Skype for Business on $strRegPool"
Start-Sleep -s 10
$Error.Clear()
try {
    Enable-CsMeetingRoom -Identity $credNewAccount.UserName -RegistrarPool $strRegPool -SipAddressType
EmailAddress
}
catch { }

if ($Error)
{
    $status["Skype for Business Account Setup"] = "Failed to setup the Skype for Business meeting room - you
can run EnableSfb.ps1 to try again."
    $Error.Clear();
}
else
{
    $status["Skype for Business Account Setup"] = "Successfully enabled account as a Skype for Business
meeting room"
}

## Now we need to assign a Skype for Business license to the account ##
# Assign a license to thes
$countryCode = (Get-CsTenant).CountryAbbreviation
$loc = Read-Host "Please enter the usage location for this device account (where the account is being used).
This is a 2-character code that is used to assign licenses (e.g. $countryCode)"
try {
    $Error.Clear()
    Set-MsolUser -UserPrincipalName $credNewAccount.UserName -UsageLocation $loc
}
catch{}
if ($Error)
{
    $status["Office 365 License"] = "Failed to assign an Office 365 license to the account"
    $Error.Clear()
}
else
{
    PrintAction "We found the following licenses available for your tenant:"
    $skus = (Get-MsolAccountSku | Where-Object { !$_.AccountSkuID.Contains("INTUNE"); })
    $i = 1
    $skus | % {
     Write-Host -NoNewline $i
     Write-Host -NoNewLine ": AccountSKUID: "
     Write-Host -NoNewLine $_.AccountSkuid
     Write-Host -NoNewLine " Active Units: "
     Write-Host -NoNewLine $_.ActiveUnits
     Write-Host -NoNewLine " Consumed Units: "
     Write-Host $_.ConsumedUnits
     $i++
    }
    $iLicenseIndex = 0;
    do
    {
        $iLicenseIndex = Read-Host 'Choose the number for the SKU you want to pick'
    } while ($iLicenseIndex -lt 1 -or $iLicenseIndex -gt $skus.Length)
    $strLicenses = $skus[$iLicenseIndex - 1].AccountSkuId

    if (![System.String]::IsNullOrEmpty($strLicenses))
    {
        try
        {
            $Error.Clear()
            Set-MsolUserLicense -UserPrincipalName $credNewAccount.UserName -AddLicenses $strLicenses
        }
```

```
        catch
        {

        }
        if ($Error)
        {
            $Error.Clear()
            $status["Office 365 License"] = "Failed to add a license to the account. Make sure you have
remaining licenses."
        }
        else
        {
            $status["Office 365 License"] = "Successfully added license to the account"
        }
    }
    else
    {
        $status["Office 365 License"] = "You opted not to install a license on this account"
    }
}


Write-Host

## Cleanup and print results ##
Cleanup
$strDisplay = $mailbox.DisplayName
$strUsr = $credNewAccount.UserName
PrintAction "Summary for creation of $strUsr ($strDisplay)"
if ($status.Count -gt 0)
{
    ForEach($k in $status.Keys)
    {
        $v = $status[$k]
        $color = "yellow"
        if ($v[0] -eq "S") { $color = "green" }
        elseif ($v[0] -eq "F")
        {
            $color = "red"
            $v += " Go to http://aka.ms/shubtshoot for help"
        }

        Write-Host -NoNewline $k -ForegroundColor $color
        Write-Host -NoNewline ": "
        Write-Host $v
    }
}
else
{
    PrintError "The account could not be created"
}
```

Account verification script

This script will validate the previously-created device account on a Surface Hub, no matter which method was used to create it. This script is basically pass/fail. If one of the test errors out, it will show a detailed error message, but if all tests pass, the end result will be a summary report. For example, you might see:

```
15 tests executed
0 failures
2 warnings
15 passed
```

Details of specific settings will not be shown.

```
# SHAccountValidate.ps1

$Error.Clear()
$ErrorActionPreference = "Stop"


# Cleans up set state such as remote powershell sessions
function Cleanup()
{
    if ($sessEx)
    {
        Remove-PSSession $sessEx
    }
    if ($sessSfb)
    {
        Remove-PSSession $sessSfb
    }
}

function PrintError($strMsg)
{
    Write-Host $strMsg -foregroundcolor "red"
}

function PrintSuccess($strMsg)
{
    Write-Host $strMsg -foregroundcolor "green"
}

function PrintAction($strMsg)
{
    Write-Host $strMsg -ForegroundColor Cyan
}


# Cleans up and prints an error message
function CleanupAndFail($strMsg)
{
    if ($strMsg)
    {
        PrintError($strMsg);
    }
    Cleanup
    exit 1
}

# Exits if there is an error set and prints the given message
function ExitIfError($strMsg)
{
    if ($Error)
    {
        CleanupAndFail($strMsg);
    }
}

$strUpn = Read-Host "What is the email address of the account you wish to validate?"
if (!$strUpn.Contains('@'))
{
    CleanupAndFail "$strUpn is not a valid email address"
}
$strExServer = Read-Host "What is your exchange server? (leave blank for online tenants)"
if ($strExServer.Equals(""))
{
    $fExIsOnline = $true
}
else
{
    $fExIsOnline = $false
```

```
}
$credEx = Get-Credential -Message "Please provide exchange user credentials"

$strRegistrarPool = Read-Host ("What is the Skype for Business registrar pool for $strUpn" + "? (leave blank
for online tenants)")
$fSfbIsOnline = $strRegistrarPool.Equals("")

$fHasOnPrem = $true
if ($fSfbIsOnline -and $fExIsOnline)
{
    do
    {
        $strHasOnPrem = (Read-Host "Do you have an on-premises Active Directory (Y/N) (No if your domain
services are hosted entirely online)").ToUpper()
    } while ($strHasOnPrem -ne "Y" -and $strHasOnPrem -ne "N")
    $fHasOnPrem = $strHasOnPrem.Equals("Y")
}

$fHasOnline = $false
if ($fSfbIsOnline -or $fExIsOnline)
{
    $fHasOnline = $true
}

if ($fSfbIsOnline)
{
    try {
        Import-Module LyncOnlineConnector
    }
    catch
    {
        CleanupAndFail "To verify Skype for Business in online tenants you need the Lync Online Connector
module from http://www.microsoft.com/download/details.aspx?id=39366"
    }
}
else
{
    $credSfb = (Get-Credential -Message "Please enter Skype for Business admin credentials")
}

if ($fHasOnline)
{
    $credSfb = $credEx
    try {
        Import-Module MSOnline
    }
    catch
    {
        CleanupAndFail "To verify accounts in online tenants you need the Azure Active Directory module for
PowerShell from https://go.microsoft.com/fwlink/p/?linkid=236297"
    }
}

PrintAction "Connecting to Exchange Powershell Session..."
[System.Management.Automation.Runspaces.AuthenticationMechanism] $authType =
[System.Management.Automation.Runspaces.AuthenticationMechanism]::Kerberos
if ($fExIsOnline)
{
    $authType = [System.Management.Automation.Runspaces.AuthenticationMechanism]::Basic
}
try
{
    $sessEx = $null
    if ($fExIsOnline)
    {
        $sessEx = New-PSSession -ConfigurationName microsoft.exchange -Credential $credEx -AllowRedirection -
Authentication $authType -ConnectionUri "https://outlook.office365.com/powershell-liveid/" -WarningAction
SilentlyContinue
    }
```

```
    }
    else
    {
        $sessEx = New-PSSession -ConfigurationName microsoft.exchange -Credential $credEx -AllowRedirection -
Authentication $authType -ConnectionUri https://$strExServer/powershell -WarningAction SilentlyContinue
    }
}
catch
{
}

if (!$sessEx)
{
    CleanupAndFail "Connecting to Exchange Powershell failed, please validate your server is accessible and
credentials are correct"
}

PrintSuccess "Connected to Exchange Powershell Session"

PrintAction "Connecting to Skype for Business Powershell Session..."

if ($fSfbIsOnline)
{
    $sessSfb = New-CsOnlineSession -Credential $credSfb
}
else
{
    $sessSfb = New-PSSession -Credential $credSfb -ConnectionURI "https://$strRegistrarPool/OcsPowershell" -
AllowRedirection -WarningAction SilentlyContinue
}

if (!$sessSfb)
{
    CleanupAndFail "Connecting to Skype for Business Powershell failed, please validate your server is
accessible and credentials are correct"
}

PrintSuccess "Connected to Skype for Business Powershell"

if ($fHasOnline)
{
    $credMsol = $null
    if ($fExIsOnline)
    {
        $credMsol = $credEx
    }
    elseif ($fSfbIsOnline)
    {
        $credMsol = $credSfb
    }
    else
    {
        CleanupAndFail "Internal error - could not determine MS Online credentials"
    }
    try
    {
        PrintAction "Connecting to Azure Active Directory Services..."
        Connect-MsolService -Credential $credMsol
        PrintSuccess "Connected to Azure Active Directory Services"
    }
    catch
    {
        # This really shouldn't happen unless there is a network error
        CleanupAndFail "Failed to connect to MSOnline"
    }
}


PrintAction "Importing remote sessions into the local session..."
try
```

```
try
{
    $importEx = Import-PSSession $sessEx -AllowClobber -WarningAction SilentlyContinue -DisableNameChecking
    $importSfb = Import-PSSession $sessSfb -AllowClobber -WarningAction SilentlyContinue -DisableNameChecking
}
catch
{
}
if (!$importEx -or !$importSfb)
{
    CleanupAndFail "Import failed"
}
PrintSuccess "Import successful"


$mailbox = $null
try
{
    $mailbox = Get-Mailbox -Identity $strUpn
}
catch
{
}

if (!$mailbox)
{
    CleanupAndFail "Account exists check failed. Unable to find the mailbox for $strUpn - please make sure the
Exchange account exists on $strExServer"
}

$exchange = $null
if (!$fExIsOnline)
{
    $exchange = Get-ExchangeServer
    if (!$exchange -or !$exchange.IsE14OrLater)
    {
        CleanupAndFail "A compatible exchange server version was not found. Please use at least exchange
2010."
    }
}


$strAlias = $mailbox.Alias
$strDisplayName = $mailbox.DisplayName

$strLinkedAccount = $strLinkedDomain = $strLinkedUser = $strLinkedServer = $null
$credLinkedDomain = $Null
if (!$fExIsOnline -and ![System.String]::IsNullOrEmpty($mailbox.LinkedMasterAccount) -and
!$mailbox.LinkedMasterAccount.EndsWith("\SELF"))
{
    $strLinkedAccount = $mailbox.LinkedMasterAccount
    $strLinkedDomain = $strLinkedAccount.substring(0,$strLinkedAccount.IndexOf('\'))
    $strLinkedUser = $strLinkedAccount.substring($strLinkedAccount.IndexOf('\') + 1)
    $strLinkedServer = Read-Host "What is the domain controller for the $strLinkedDomain"
    $credLinkedDomain = (Get-Credential -Message "Please provide credentials for $strLinkedDomain")
}







Write-Host
Write-Host
Write-Host
PrintAction "Performing verification checks on $strDisplayName..."
$Global:iTotalFailures = 0
$global:iTotalWarnings = 0
```

```
$Global:iTotalPasses = 0

function Validate()
{
    Param(
        [string]$Test,
        [bool]  $Condition,
        [string]$FailureMsg,
        [switch]$WarningOnly
    )

    Write-Host -NoNewline -ForegroundColor White $Test.PadRight(100,'.')
    if ($Condition)
    {
        Write-Host -ForegroundColor Green "Passed"
        $global:iTotalPasses++
    }
    else
    {
        if ($WarningOnly)
        {
            Write-Host -ForegroundColor Yellow ("Warning: "+$FailureMsg)
            $global:iTotalWarnings++
        }
        else
        {
            Write-Host -ForegroundColor Red ("Failed: "+$FailureMsg)
            $global:iTotalFailures++
        }
    }
}

## Exchange ##

Validate -WarningOnly -Test "The mailbox $strUpn is enabled as a room account" -Condition
($mailbox.RoomMailboxAccountEnabled -eq $True) -FailureMsg "RoomMailboxEnabled - without a device account, the
Surface Hub will not be able to use various key features."
$calendarProcessing = Get-CalendarProcessing -Identity $strUpn -WarningAction SilentlyContinue -ErrorAction
SilentlyContinue
Validate -Test "The mailbox $strUpn is configured to accept meeting requests" -Condition ($calendarProcessing
-ne $null -and $calendarProcessing.AutomateProcessing -eq 'AutoAccept') -FailureMsg "AutomateProcessing - the
Surface Hub will not be able to send mail or sync its calendar."
Validate -WarningOnly -Test "The mailbox $strUpn will not delete meeting comments" -Condition
($calendarProcessing -ne $null -and !$calendarProcessing.DeleteComments) -FailureMsg "DeleteComments - the
Surface Hub may be missing some meeting information on the welcome screen and Skype."
Validate -WarningOnly -Test "The mailbox $strUpn keeps private meetings private" -Condition
($calendarProcessing -ne $null -and !$calendarProcessing.RemovePrivateProperty) -FailureMsg
"RemovePrivateProperty - the Surface Hub will make show private meetings."
Validate -Test "The mailbox $strUpn keeps meeting subjects" -Condition ($calendarProcessing -ne $null -and
!$calendarProcessing.DeleteSubject) -FailureMsg "DeleteSubject - the Surface Hub will not keep meeting subject
information."
Validate -WarningOnly -Test "The mailbox $strUpn does not prepend meeting organizers to subjects" -Condition
($calendarProcessing -ne $null -and !$calendarProcessing.AddOrganizerToSubject) -FailureMsg
"AddOrganizerToSubject - the Surface Hub will not display meeting subjects as intended."

if ($fExIsOnline)
{
    #No online specifics
}
else
{
    #No onprem specifics
}

#ActiveSync
$casMailbox = Get-Casmailbox $strUpn -WarningAction SilentlyContinue -ErrorAction SilentlyContinue
Validate -Test "The mailbox $strUpn has a mailbox policy" -Condition ($casMailbox -ne $null) -FailureMsg
"PasswordEnabled - unable to find policy - the Surface Hub will not be able to send mail or sync its
calendar."
```

```
if ($casMailbox)
{
    $policy = $null
    if ($fExIsOnline -or $exchange.IsE15OrLater)
    {
        $strPolicy = $casMailbox.ActiveSyncMailboxPolicy
        $policy = Get-MobileDeviceMailboxPolicy -Identity $strPolicy -WarningAction SilentlyContinue -
ErrorAction SilentlyContinue
        Validate -Test "The policy $strPolicy does not require a device password" -Condition
($policy.PasswordEnabled -ne $True) -FailureMsg "PasswordEnabled - policy requires a device password - the
Surface Hub will not be able to send mail or sync its calendar."
    }
    else
    {
        $strPolicy = $casMailbox.ActiveSyncMailboxPolicy
        $policy = Get-ActiveSyncMailboxPolicy -Identity $strPolicy -WarningAction SilentlyContinue -
ErrorAction SilentlyContinue
        Validate -Test "The policy $strPolicy does not require a device password" -Condition
($policy.PasswordEnabled -ne $True) -FailureMsg "PasswordEnabled - policy requires a device password - the
Surface Hub will not be able to send mail or sync its calendar."
    }

    if ($policy -ne $null)
    {
        Validate -Test "The policy $strPolicy allows non-provisionable devices" -Condition
($policy.AllowNonProvisionableDevices -eq $null -or $policy.AllowNonProvisionableDevices -eq $true) -
FailureMsg "AllowNonProvisionableDevices - policy will not allow the SurfaceHub to sync"
    }

}


# Check the default access level
$orgSettings = Get-ActiveSyncOrganizationSettings
$strDefaultAccessLevel = $orgSettings.DefaultAccessLevel
Validate -Test "ActiveSync devices are allowed" -Condition ($strDefaultAccessLevel -eq 'Allow') -FailureMsg
"DeviceType Windows Mail is accessible - devices are not allowed by default - the surface hub will not be able
to send mail or sync its calendar."

# Check if there exists a device access rule that bans the device type Windows Mail
$blockingRules = Get-ActiveSyncDeviceAccessRule | where {($_.AccessLevel -eq 'Block' -or $_.AccessLevel -eq
'Quarantine') -and $_.Characteristic -eq 'DeviceType'-and $_.QueryString -eq 'WindowsMail'}
Validate -Test "Windows mail devices are not blocked or quarantined" -Condition ($blockingRules -eq $null -or
$blockingRules.Length -eq 0) -FailureMsg "DeviceType Windows Mail is accessible - devices are blocked or
quaratined - the surface hub will not be able to send mail or sync its calendar."

## End Exchange ##



## SfB ##
$strLyncIdentity = $null
if ($fSfbIsOnline)
{
    $strLyncIdentity = $strUpn
}
else
{
    $strLyncIdentity = $strAlias
}

$lyncAccount = $null
try {
    $lyncAccount = Get-CsMeetingRoom -Identity $strLyncIdentity -WarningAction SilentlyContinue -ErrorAction
SilentlyContinue
} catch {
    try {
        $lyncAccount = Get-CsUser -Identity $strLyncIdentity -WarningAction SilentlyContinue -ErrorAction
SilentlyContinue
```

```
    } catch { }
}
Validate -Test "There is a Lync or Skype for Business account for $strLyncIdentity" -Condition ($lyncAccount -
ne $null -and $lyncAccount.Enabled) -FailureMsg "SfB Enabled - there is no Skype for Business account -
meetings will not support Skype for Business"
if ($lyncAccount)
{
    Validate -Test "The meeting room has a SIP address" -Condition (!
[System.String]::IsNullOrEmpty($lyncAccount.SipAddress)) -FailureMsg "SfB Enabled - there is no SIP Address -
the device account cannot be used to sign into Skype for Business."
}
## End SFB ##


if ($fHasOnline)
{
    #License validation and password expiry
    $accountOnline = Get-MsolUser -UserPrincipalName $strUpn -WarningAction SilentlyContinue -ErrorAction
SilentlyContinue
    Validate -Test "There is an online user account for $strUpn" -Condition ($accountOnline -ne $null) -
FailureMsg "Could not find a Microsoft Online account for this user even though some services are online"
    if ($accountOnline)
    {
        Validate -Test "The password for $strUpn will not expire" -Condition
($accountOnline.PasswordNeverExpires -eq $True) -FailureMsg "PasswordNeverExpires - the admin will need to
update the device account's password on the Surface Hub when it expires."
        if ($fIsSfbOnline -and !$fIsExOnline)
        {
            $strLicenseFailureMsg = "Has O365 license - The devices will not be able to use Skype for Business
services."
        }
        elseif ($fIsExOnline -and !$fIsSfbOnline)
        {
            $strLicenseFailureMsg = "Has O365 license - The devices will not be able to use Exchange Online
services."
        }
        else
        {
            $strLicenseFailureMsg = "Has O365 license - The devices will not be able to use Skype for Business
or Exchange Online services."
        }
        Validate -Test "$strUpn is licensed" -Condition ($accountOnline.IsLicensed -eq $True) -FailureMsg
$strLicenseFailureMsg

        Validate -Test "$strUpn is allowed to sign in" -Condition ($accountOnline.BlockCredential -ne $True) -
FailureMsg "BlockCredential - This user is not allowed to sign in."
    }
}

#If there is an on-prem component, we can get the authorative AD user from mailbox
if ($fHasOnPrem)
{
    $accountOnPrem = $null
    if ($strLinkedAccount)
    {
        $accountOnPrem = Get-AdUser $strLinkedUser -server $strLinkedServer -credential $credLinkedDomain -
properties PasswordNeverExpires -WarningAction SilentlyContinue -ErrorAction SilentlyContinue
    }
    else
    {
        #AD User enabled validation
        $accountOnPrem = Get-AdUser $strAlias -properties PasswordNeverExpires -WarningAction SilentlyContinue
-ErrorAction SilentlyContinue
    }
    $strOnPremUpn = $accountOnPrem.UserPrincipalName
    Validate -Test "There is a user account for $strOnPremUpn" -Condition ($accountOnprem -ne $null) -
FailureMsg "Could not find an Active Directory account for this user"
    if ($accountOnPrem)
    {
```

```
        Validate -WarningOnly -Test "The password for $strOnPremUpn will not expire" -Condition
($accountOnprem.PasswordNeverExpires -eq $True) -FailureMsg "PasswordNeverExpires - the admin will need to
update the device account's password on the Surface Hub when it expires."
        Validate -Test "$strOnPremUpn is enabled" -Condition $accountOnPrem.Enabled -FailureMsg
"AccountEnabled - this device account will not sign in"
    }
}


$global:iTotalTests = ($global:iTotalFailures + $global:iTotalPasses + $global:iTotalWarnings)

Write-Host -NoNewline $global:iTotalTests "tests executed: "
Write-Host -NoNewline -ForegroundColor Red $Global:iTotalFailures "failures "
Write-Host -NoNewline -ForegroundColor Yellow $Global:iTotalWarnings "warnings "
Write-Host -ForegroundColor Green $Global:iTotalPasses "passes "

Cleanup
```

Enable Skype for Business

This script will enable Skype for Business on a device account. Use it only if Skype for Business wasn't previously enabled during account creation.

```
## This script performs only the Enable for Skype for Business step on an account. It should only be run if
this step failed in SHAccountCreate and the other steps have been completed ##
# EnableSfb.ps1

$Error.Clear()
$ErrorActionPreference = "Stop"

# Cleans up set state such as remote powershell sessions
function Cleanup()
{
    if ($sessCS)
    {
        Remove-PSSession $sessCS
    }
}

function PrintError($strMsg)
{
    Write-Host $strMsg -foregroundcolor "red"
}

function PrintSuccess($strMsg)
{
    Write-Host $strMsg -foregroundcolor "green"
}

# Cleans up and prints an error message
function CleanupAndFail($strMsg)
{
    if ($strMsg)
    {
        PrintError($strMsg);
    }
    Cleanup
    exit 1
}

# Exits if there is an error set and prints the given message
function ExitIfError($strMsg)
{
    if ($Error)
    {
        CleanupAndFail($strMsg);
```

```
        CleanupAndFail($strMsg);
    }
}

## Check dependencies ##

$input = Read-Host "Is the account you wish to enable part of an online environment (enter O) or on-premises
environment (enter P)"
if ($input -eq "P")
{
    $online = $false
}
elseif ($input -eq "O")
{
    $online = $true
}
else
{
    CleanupAndFail "Invalid selection"
}
if ($online)
{
    try {
        Import-Module LyncOnlineConnector
    }
    catch
    {
        PrintError "Some dependencies are missing"
        PrintError "Please install the Windows PowerShell Module for Lync Online. For more information go to
http://www.microsoft.com/download/details.aspx?id=39366"
        PrintError "Please install the Azure Active Directory module for PowerShell from
https://go.microsoft.com/fwlink/p/?linkid=236297"
        CleanupAndFail
    }
}
else
{
    $strRegPool = Read-Host "Enter the FQDN of your Skype for Business Registrar Pool"
}


## Collect account data ##
Write-Host "-----------  Enter info for the account to enable  -----------." -foregroundcolor "magenta"
$strRoomUri=Read-Host 'Please enter the UPN of the account you are enabling (e.g.
confroom@surfacehub.microsoft.com)'

if ([System.String]::IsNullOrEmpty($strRoomUri))
{
    CleanupAndFail "Please enter all of the requested data to continue."
    exit 1
}
Write-Host "------------------------------------------------------------------." -foregroundcolor "magenta"



## Sign in to remote powershell for exchange and lync online ##
Write-Host "`n-----------------  Establishing connection  -----------------." -foregroundcolor "magenta"
$credAdmin=Get-Credential -Message "Enter credentials of a Skype for Business admin"
if (!$credadmin)
{
    CleanupAndFail("Valid admin credentials are required to create and prepare the account.");
}
Write-Host "Connecting to remote sessions. This can occasionally take a while - please do not enter input..."

try
{
    if ($online)
    {
        $sessCS = New-CsOnlineSession -Credential $credAdmin
    }
```

```
    }
    else
    {
        $sessCS = New-PSSession -Credential $credAdmin -ConnectionURI "https://$strRegPool/OcsPowershell" -
AllowRedirection -WarningAction SilentlyContinue
    }
}
catch
{
    CleanupAndFail("Failed to connect to Skype for Business server. Please check your credentials and try
again. Error message: $_")
}

Import-PSSession $sessCS -AllowClobber

# In case there was any uncaught errors
ExitIfError("Remote connection failed. Please check your credentials and try again.")
Write-Host "-------------------------------------------------------------" -foregroundcolor "magenta"

# Getting registrar pool
if ($online)
{
    try {
        $strRegPool = $null;
        $strRegPool = (Get-CsTenant).TenantPoolExtension
    } catch {}
    if ($Error)
    {
        $Error.Clear();
        $strRegPool = "";
        Write-Host "We failed to lookup your Skype for Business Registrar Pool, but you can still enter it
manually"
    }
    else
    {
        $strRegPool = $strRegPool[0].Substring($strRegPool[0].IndexOf(':') + 1)
    }
}


$Error.Clear()
try {
    Enable-CsMeetingRoom -Identity $strRoomUri -RegistrarPool $strRegPool -SipAddressType EmailAddress
}
catch {}

ExitIfError("Failed to setup Skype for Business meeting room")

PrintSuccess "Successfully enabled $strRoomUri as a Skype for Business meeting room"

Cleanup
```

Useful cmdlets

Creating a Surface Hub-compatible ActiveSync policy

For Surface Hub to use Exchange services, a device account configured with a compatible ActiveSync policy must be provisioned on the device. This policy has the following requirements:

```
PasswordEnabled == 0
```

In the following cmdlets, `$strPolicy` is the name of the ActiveSync policy, and `$strRoomUpn` is the UPN of the device account you want to apply the policy to.

Note that in order to run the cmdlets, you need to set up a remote PowerShell session and:

- Your admin account must be remote-PowerShell-enabled. This allows the admin to use the PowerShell cmdlets that are needed by the script. (This permission can be set using `set-user $admin -RemotePowerShellEnabled $true`)
- Your admin account must have the "Reset Password" role if you plan to run the creation scripts. This allows the admin to change the password of the account, which is needed for the script. The Reset Password Role can be enabled using the Exchange Admin Center.

Create the policy.

```
# Create new policy with PasswordEnabled == false
New-MobileDeviceMailboxPolicy -Name $strPolicy -PasswordEnabled $false –AllowNonProvisionableDevices $true
```

To apply the policy, the mailbox cannot be a room type, so it has to be converted into a user first.

```
# Convert user to regular type
Set-Mailbox $strRoomUpn -Type Regular
# Set policy for account
Set-CASMailbox $strRoomUpn -ActiveSyncMailboxPolicy $strPolicy
```

Now the device account just needs to be converted back into a room type.

```
# Convert back to room mailbox
Set-Mailbox $strRoomUpn -Type Room
```

Allowing device IDs for ActiveSync

To allow an account `$strRoomUpn`, run the following command:

```
Set-CASMailbox –Identity $strRoomUpn –ActiveSyncAllowedDeviceIDs “<ID>”
```

To find a device's ID, run:

```
Get-ActiveSyncDevice -Mailbox $strRoomUpn
```

This retrieves device information for every device that the account has been provisioned on, Including the `DeviceId` property.

Auto-accepting and declining meeting requests

For a device account to automatically accept or decline meeting requests based on its availability, the **AutomateProcessing** attribute must be set to **AutoAccept**. This is recommended as to prevent overlapping meetings.

```
Set-CalendarProcessing $strRoomUpn -AutomateProcessing AutoAccept
```

Accepting external meeting requests

For a device account to accept external meeting requests (a meeting request from an account not in the same tenant/domain), the device account must be set to allow processing of external meeting requests. Once set, the device account will automatically accept or decline meeting requests from external accounts as well as local accounts.

> **Note** If the **AutomateProcessing** attribute is not set to **AutoAccept**, then setting this will have no effect.

```
Set-CalendarProcessing $strRoomUpn -ProcessExternalMeetingMessages $true
```

Useful downloads for Microsoft Surface Hub

5/4/2017 • 2 min to read • Edit Online

This topic provides links to useful Surface Hub documents, such as product datasheets, the site readiness guide, and user's guide.

LINK	DESCRIPTION
Surface Hub Site Readiness Guide (PDF)	Make sure your site is ready for Surface Hub, including structural and power requirements, and get technical specs for Surface Hub. Watch the video (opens in a pop-up media player)
Surface Hub Setup Guide (English, French, Spanish) (PDF)	Get a quick overview of how to set up the environment for your new Surface Hub.
Surface Hub Quick Reference Guide (PDF)	Use this quick reference guide to get information about key features and functions of the Surface Hub.
Surface Hub User Guide (PDF)	Learn how to use Surface Hub in scheduled or ad-hoc meetings. Invite remote participants, use the built-in tools, save data from your meeting, and more.
Surface Hub Replacement PC Drivers	The Surface Hub Replacement PC driver set is available for those customers who have chosen to disable the Surface Hub's internal PC and use an external computer with their 84" or 55" Surface Hub. This download is meant to be used with the Surface Hub Admin Guide , which contains further details on configuring a Surface Hub Replacement PC.
Surface Hub SSD Replacement Guide (PDF)	Learn how to replace the solid state drive (SSD) for the 55- and 84-inch Surface Hub.
Microsoft Surface Hub Rollout and Adoption Success Kit (ZIP)	Best practices for generating awareness and implementing change management to maximize adoption, usage, and benefits of Microsoft Surface Hub. The Rollout and Adoption Success Kit zip file includes the Rollout and Adoption Success Kit detailed document, Surface Hub presentation, demo guidance, awareness graphics, and more.
Unpacking Guide for 84-inch Surface Hub (PDF)	Learn how to unpack your 84-inch Surface Hub efficiently and safely. Watch the video (opens in a pop-up media player)
Unpacking Guide for 55-inch Surface Hub (PDF)	Learn how to unpack your 55-inch Surface Hub efficiently and safely. Watch the video (opens in a pop-up media player)
Wall Mounting and Assembly Guide (PDF)	Detailed instructions on how to safely and securely assemble the wall brackets, and how to mount your Surface Hub onto them. Watch the video (opens in a pop-up media player)

LINK	DESCRIPTION
Floor-Supported Mounting and Assembly Guide (PDF)	Detailed instructions on how to safely and securely assemble the floor-supported brackets, and how to mount your Surface Hub onto them. Watch the video (opens in a pop-up media player)
Rolling Stand Mounting and Assembly Guide (PDF)	Detailed instructions on how to safely and securely assemble the rolling stand, and how to mount your Surface Hub onto it. Watch the video (opens in a pop-up media player)
Mounts and Stands Datasheet (PDF)	Specifications and prices for all Surface Hub add-on stands and mounts that turn your workspace into a Surface Hub workspace.
Surface Hub Stand and Wall Mount Specifications (PDF)	Illustrated specifications for the 55″ and 84″ Surface Hub rolling stands, wall mounts, and floor-supported wall mounts.
Surface Hub Onsite Installation and Onsite Repair/Exchange Services FAQ (PDF)	Get answers to the most common questions about Surface Hub onsite service offerings and delivery.

Differences between Surface Hub and Windows 10 Enterprise

5/4/2017 • 6 min to read • Edit Online

The Surface Hub operating system, Windows 10 Team, is based on Windows 10 Enterprise, providing rich support for enterprise management, security, and other features. However, there are important differences between them. While the Enterprise edition is designed for PCs, Windows 10 Team is designed from the ground up for large screens and meeting rooms. When you evaluate security and management requirements for Surface Hub, it's best to consider it as a new operating system. This article is designed to help highlight the key differences between Windows 10 Team on Surface Hub and Windows 10 Enterprise, and what the differences mean for your organization.

User interface

Shell (OS user interface)

The Surface Hub's shell is designed from the ground up to be large screen and touch optimized. It doesn't use the same shell as Windows 10 Enterprise.

Organization policies that this may affect:
Settings related to controls in the Windows 10 Enterprise shell don't apply for Surface Hub.

Lock screen and screensaver

Surface Hub doesn't have a lock screen or a screen saver, but it has a similar feature called the welcome screen. The welcome screen shows scheduled meetings from the device account's calendar, and easy entry points to the Surface Hub's top apps - Skype for Business, Whiteboard, and Connect.

Organization policies that this may affect:
Settings for lock screen, screen timeout, and screen saver don't apply for Surface Hub.

User logon

Surface Hub is designed to be used in communal spaces, such as meeting rooms. Unlike Windows PCs, anyone can walk up and use a Surface Hub without logging on. The system always runs as a local, auto logged-in, low-privilege user. It doesn't support logging in any additional users - including admin users.

> **NOTE**
> Surface Hub supports signing in to Microsoft Edge and other apps. However, these credentials are deleted when users press **I'm done**.

Organization policies that this may affect:
Generally, Surface Hub uses lockdown features rather than user access control to enforce security. Policies related to password requirements, interactive logon, user accounts, and access control don't apply for Surface Hub.

Saving and browsing files

Users have access to a limited set of directories on the Surface Hub:

- Music
- Videos
- Documents

- Pictures
- Downloads

Files saved locally in these directories are deleted when users press **I'm done**. To save content created during a meeting, users should save files to a USB drive or to OneDrive.

Organization policies that this may affect:
Policies related to access permissions and ownership of files and folders don't apply for Surface Hub. Users can't browse and save files to system directories and network folders.

Applications

Default applications

With few exceptions, the default Universal Windows Platform (UWP) apps on Surface Hub are also available on Windows 10 PCs.

UWP apps pre-installed on Surface Hub:

- Alarms & Clock
- Calculator
- Connect
- Excel Mobile
- Feedback Hub
- File Explorer*
- Get Started
- Maps
- Microsoft Edge
- Microsoft Power BI
- OneDrive
- Photos
- PowerPoint Mobile
- Settings*
- Skype for Business*
- Store
- Whiteboard*
- Word Mobile

Apps with an asterisk (*) are unique to Surface Hub

Organization policies that this may affect:
Use guidelines for Windows 10 Enterprise to determine the features and network requirements for default apps on the Surface Hub.

Installing apps, drivers, and services

To help preserve the appliance-like nature of the device, Surface Hub only supports installing Universal Windows Platform (UWP) apps, and does not support installing classic Win32 apps, services and drivers. Furthermore, only admins have access to install UWP apps.

Organization policies that this may affect:
Employees can only use the apps that have been installed by admins, helping mitigate against unintended use. Surface Hub doesn't support installing Win32 agents required by most traditional PC management and monitoring tools.

Security and lockdown

For Surface Hub to be used in communal spaces, such as meeting rooms, its custom OS implements many of the security and lockdown features available in Windows 10.

Surface Hub implements these Windows 10 security features:

- UEFI Secure Boot
- User Mode Code Integrity (UMCI) with Device Guard
- Application restriction policies using AppLocker
- BitLocker Drive Encryption
- Trusted Platform Module (TPM)
- Windows Defender
- User Account Control (UAC) for access to the Settings app

These Surface Hub features provide additional security:

- Custom UEFI firmware
- Custom shell and Start menu limits device to meeting functions
- Custom File Explorer only grants access to files and folders under My Documents
- Custom Settings app only allows admins to modify device settings
- Downloading advanced Plug and Play drivers is disabled

Organization policies that this may affect:
Consider these features when performing your security assessment for Surface Hub.

Management

Device settings

Device settings can be configured through the Settings app. The Settings app is customized for Surface Hub, but also contains many familiar settings from Windows 10 Desktop. A User Accounts Control (UAC) prompt appears when opening up the Settings app to verify the admin's credentials, but this does not log in the admin.

Organization policies that this may affect:
Employees can use the Surface Hub for meetings, but cannot modify any device settings. In addition to lockdown features, this ensures that employees only use the device for meeting functions.

Administrative features

The administrative features in Windows 10 Enterprise, such as the Microsoft Management Console, Run, Command Prompt, PowerShell, registry editor, event viewer, and task manager are not supported on Surface Hub. The Settings app contains all of the administrative features locally available on Surface Hub.

Organization policies that this may affect:
Surface Hubs are not managed like traditional PCs. Use MDM to configure settings and OMS to monitor your Surface Hub.

Remote management and monitoring

Surface Hub supports remote management through mobile device management (MDM), and monitoring through Operations Management Suite (OMS).

Organization policies that this may affect:
Surface Hub doesn't support installing Win32 agents required by most traditional PC management and monitoring tools, such as System Center Operations Manager.

Group policy

Surface Hub does not support group policy, including auditing. Instead, use MDM to apply policies to your Surface Hub. For more information about MDM, see Manage settings with an MDM provider.

Organization policies that this may affect:
Use MDM to manage Surface Hub rather than group policy.

Remote assistance

Surface Hub does not support remote assistance.

Organization policies that this may affect:
Policies related to remote assistance don't apply for Surface Hub.

Network

Domain join and Azure Active Directory (Azure AD) join

Surface Hub uses domain join and Azure AD join primarily to provide a directory-backed admin group. Users can't log in with a domain account. For more information, see Admin group management.

Organization policies that this may affect:
Group policies are not applied when a Surface Hub is joined to your domain. Policies related to domain membership don't apply for Surface Hub.

Accessing domain resources

Users can sign in to Microsoft Edge to access intranet sites and online resources (such as Office 365). If your Surface Hub is configured with a device account, the system uses it to access Exchange and Skype for Business. However, Surface Hub doesn't support accessing domain resources such as file shares and printers.

Organization policies that this may affect:
Policies related to accessing domain objects don't apply for Surface Hub.

Telemetry

The Surface Hub OS uses the Windows 10 Connected User Experience and Telemetry component to gather and transmit telemetry data. For more information, see Configure Windows telemetry in your organization.

Organization policies that this may affect:
Configure telemetry levels for Surface Hub in the same way as you do for Windows 10 Enterprise.

How Surface Hub addresses Wi-Fi Direct security issues

5/4/2017 • 11 min to read • Edit Online

Microsoft Surface Hub is an all-in-one productivity device that enables teams to better brainstorm, collaborate, and share ideas. Surface Hub relies on Miracast for wireless projection by using Wi-Fi Direct.

This topic provides guidance on Wi-Fi Direct security vulnerabilities, how Surface Hub has addressed those risks, and how Surface Hub administrators can configure the device for the highest level of security. This hardening information will help customers with high security requirements understand how best to protect their Surface Hub connected networks and data in transit.

The intended audiences for this topic include IT and network administrators interested in deploying Microsoft Surface Hub in their corporate environment with optimal security settings.

Overview

Microsoft Surface Hub's security depends extensively on Wi-Fi Direct / Miracast and the associated 802.11, Wi-Fi Protected Access (WPA2), and Wireless Protected Setup (WPS) standards. Since the device only supports WPS (as opposed to WPA2 Pre-Shared Key (PSK) or WPA2 Enterprise), issues traditionally associated with 802.11 encryption are simplified by design.

It is important to note Surface Hub operates on par with the field of Miracast receivers, meaning that it is protected from, and vulnerable to, a similar set of exploits as all WPS-based wireless network devices. But Surface Hub's implementation of WPS has extra precautions built in, and its internal architecture helps prevent an attacker – even after compromising the Wi-Fi Direct / Miracast layer – to move past the network interface onto other attack surfaces and connected enterprise networks see Wi-Fi Direct vulnerabilities and how Surface Hub addresses them.

Wi-Fi Direct background

Miracast is part of the Wi-Fi Display standard, which itself is supported by the Wi-Fi Direct protocol. These standards are supported in modern mobile devices for screen sharing and collaboration.

Wi-Fi Direct or Wi-Fi "Peer to Peer" (P2P) is a standard released by the Wi-Fi Alliance for "Ad-Hoc" networks. This allows supported devices to communicate directly and create groups of networks without requiring a traditional Wi-Fi Access Point or an Internet connection.

Security for Wi-Fi Direct is provided by WPA2 using the WPS standard. Authentication mechanism for devices can be a numerical pin (WPS-PIN), a physical or virtual Push Button (WPS-PBC), or an out-of-band message such as Near Field Communication (WPS-OOO). The Microsoft Surface Hub supports both Push Button (which is the default) and PIN methods.

In Wi-Fi Direct, groups are created as either "persistent," allowing for automatic reconnection using stored key material, or "temporary," where devices cannot re-authenticate without user intervention or action. Wi-Fi Direct groups will typically determine a Group Owner (GO) through a negotiation protocol, which mimics the "station" or "Access Point" functionality for the established Wi-Fi Direct Group. This Wi-Fi Direct GO provides authentication (via an "Internal Registrar"), and facilitate upstream network connections. For Surface Hub, this GO negotiation does not take place, as the network only operates in "autonomous" mode, where Surface Hub is always the Group Owner. Finally, Surface Hub does not and will not join other Wi-Fi Direct networks itself as a client.

Wi-Fi Direct vulnerabilities and how Surface Hub addresses them

Vulnerabilities and attacks in the Wi-Fi Direct invitation, broadcast, and discovery process: Wi-Fi Direct / Miracast attacks may target weaknesses in the group establishment, peer discovery, device broadcast, or invitation processes.

WI-FI DIRECT VULNERABILITY	SURFACE HUB MITIGATION
The discovery process may remain active for an extended period of time, which could allow Invitations and connections to be established without the intent of the device owner.	Surface Hub only operates as the Group Owner (GO), which does not perform the client Discovery or GO negotiation process. Broadcast can be turned off by fully disabling wireless projection.
Invitation and discovery using PBC allows an unauthenticated attacker to perform repeated connection attempts or unauthenticated connections are automatically accepted.	By requiring WPS PIN security, Administrators can reduce the potential for such unauthorized connections or "Invitation bombs" (where invitations are repeatedly sent until a user mistakenly accepts one).

Wi-Fi Protected Setup (WPS) Push Button Connect (PBC) vs PIN Entry: Public weaknesses have been demonstrated in WPS-PIN method design and implementation, other vulnerabilities exist within WPS-PBC involving active attacks against a protocol designed for one time use.

WI-FI DIRECT VULNERABILITY	SURFACE HUB MITIGATION
WPS-PBC is vulnerable to active attackers. As stated within the WPS specification: "The PBC method has zero bits of entropy and only protects against passive eavesdropping attacks. PBC protects against eavesdropping attacks and takes measures to prevent a device from joining a network that was not selected by the device owner. The absence of authentication, however, means that PBC does not protect against active attack". Attackers can use selective wireless jamming or other potential denial-of-service vulnerabilities in order to trigger an unintended Wi-Fi Direct GO or connection. Additionally, an active attacker, with only physical proximity, can repeatedly teardown any Wi-Fi Direct group and attempt the described attack until it is successful.	Enable WPS-PIN security within Surface Hub's configuration. As discussed within the Wi-Fi WPS specification: "The PBC method should only be used if no PIN-capable Registrar is available and the WLAN user is willing to accept the risks associated with PBC".
WPS-PIN implementations can be brute-forced using a Vulnerability within the WPS standard. Due to the design of split PIN verification, a number of implementation vulnerabilities occurred in the past several years across a wide range of Wi-Fi hardware manufacturers. In 2011 two researchers (Stefan Viehböck and Craig Heffner) released information on this vulnerability and tools such as "Reaver" as a proof of concept.	The Microsoft implementation of WPS within Surface Hub changes the pin every 30 seconds. In order to crack the pin, an attacker must work through the entire exploit in less than 30 seconds. Given the current state of tools and research in this area, a brute-force pin-cracking attack through WPS is unlikely.
WPS-PIN can be cracked using an offline attack due to weak initial key (E-S1,E S2) entropy. In 2014, Dominique Bongard discussed a "Pixie Dust" attack where poor initial randomness for the pseudo random number generator (PRNG) within the wireless device lead to the ability to perform an offline brute-force attack.	The Microsoft implementation of WPS within Surface Hub is not susceptible to this offline PIN brute-force attack. The WPS-PIN is randomized for each connection.

Unintended exposure of network services: Network daemons intended for Ethernet or WLAN services may be accidentally exposed due to misconfiguration (such as binding to "all"/0.0.0.0 interfaces), a poorly configured device firewall, or missing firewall rules altogether.

WI-FI DIRECT VULNERABILITY	SURFACE HUB MITIGATION
Misconfiguration binds a vulnerable or unauthenticated network service to "all" interfaces, which includes the Wi-Fi Direct interface. This potentially exposes services not intended to be accessible to Wi-Fi Direct clients, which may be weakly or automatically authenticated.	Within Surface Hub, the default firewall rules only permit the required TCP and UDP network ports and by default deny all inbound connections. Strong authentication can be configured by enabling the WPS-PIN mode.

Bridging Wi-Fi Direct and other wired or wireless networks: While network bridging between WLAN or Ethernet networks is a violation of the Wi-Fi Direct specification, such a bridge or misconfiguration may effectively lower or remove wireless access controls for the internal corporate network.

WI-FI DIRECT VULNERABILITY	SURFACE HUB MITIGATION
Wi-Fi Direct devices could allow unauthenticated or poorly authenticated access to bridged network connections. This may allow Wi-Fi Direct networks to route traffic to internal Ethernet LAN or other infrastructure or enterprise WLAN networks in violation of existing IT security protocols.	Surface Hub cannot be configured to bridge Wireless interfaces or allow routing between disparate networks. The default firewall rules add defense in depth to any such routing or bridge connections.

The use of Wi-Fi Direct “legacy” mode: Exposure to unintended networks or devices when operating in “legacy” mode may present a risk. Device spoofing or unintended connections could occur if WPS-PIN is not enabled.

WI-FI DIRECT VULNERABILITY	SURFACE HUB MITIGATION
By supporting both Wi-Fi Direct and 802.11 infrastructure clients, the system is operating in a "legacy" support mode. This may expose the connection setup phase indefinitely, allowing for groups to be joined or devices invited to connect well after their intended setup phase terminates.	Surface Hub does not support Wi-Fi Direct legacy clients. Only Wi-Fi Direct connections can be made to Surface Hub even when WPS-PIN mode is enabled.

Wi-Fi Direct GO negotiation during connection setup: The Group Owner within Wi-Fi Direct is analogous to the “Access Point” in a traditional 802.11 wireless network. The negotiation can be gamed by a malicious device.

WI-FI DIRECT VULNERABILITY	SURFACE HUB MITIGATION
If groups are dynamically established or if the Wi-Fi Direct device can be made to join new groups, the Group Owner (GO) negotiation can be won by a malicious device that always specifies the max Group Owner "intent" value of 15. (Unless such device is configured to always be a Group Owner, in which case the connection fails.)	Surface Hub takes advantage of Wi-Fi Direct "Autonomous mode", which skips the GO negotiation phase of the connection setup. Surface Hub is always the Group Owner.

Unintended or malicious Wi-Fi deauthentication: Wi-Fi deauthentication is an age-old attack that can be used by a physically local attacker to expedite information leaks against the connection setup process, trigger new four-way handshakes, target Wi-Fi Direct WPS-PBC for active attack, or create denial-of-service attacks.

WI-FI DIRECT VULNERABILITY	SURFACE HUB MITIGATION
Deauthentication packets can be sent by an unauthenticated attacker to cause the station to re-authenticate and sniff the resulting handshake. Cryptographic or brute-force attacks can be attempted on the resulting handshake. Mitigations for these attack include: enforcing length and complexity policies for pre-shared keys; configuring the Access Point (if applicable) to detect malicious levels of deauthentication packets; and using WPS to automatically generate strong keys. In PBC mode the user is interacting with a physical or virtual button to allow arbitrary device association. This process should happen only at setup within a small window, once the button is automatically "pushed", the device will accept any station associating via a canonical PIN value (all zeros). Deauthentication can force a repeated setup process.	The current Surface Hub design uses WPS in PIN or PBC mode. No PSK configuration is permitted, helping enforce the generation of strong keys. It is recommended to enable WPS-PIN.
Beyond denial-of-service attacks, deauthentication packets can also be used to trigger a reconnect which re-opens the window of opportunity for active attacks against WPS-PBC.	Enable WPS-PIN security within Surface Hub's configuration.

Basic wireless information disclosure: Wireless networks, 802.11 or otherwise, are inherently sources of information disclosure. Although the information is largely connection or device metadata, it remains an accepted risk for any 802.11 administrator. Wi-Fi Direct with device authentication via WPS-PIN effectively reveals the same information as a PSK or Enterprise 802.11 network.

WI-FI DIRECT VULNERABILITY	SURFACE HUB MITIGATION
During broadcast, connection setup, or even with already encrypted connections, basic information about the devices and packet sizes is wirelessly transmitted. At a basic level, a local attacker within wireless range can determine the names of wireless devices, the MAC addresses of communicating equipment, and possibly other details such as the version of the wireless stack, packet sizes, or the configured Access Point or Group Owner options by examining the relevant 802.11 Information Elements.	The Wi-Fi Direct network employed by Surface Hub cannot be further protected from metadata leaks, in the same way 802.11 Enterprise or PSK wireless networks also leak such metadata. Physical security and removing potential threats from the wireless proximity can be used to reduce any potential information leaks.

Wireless evil twin or spoofing attacks: Spoofing the wireless name is a trivial and known exploit for a physically local attacker in order to lure unsuspecting or mistaken users to connect.

WI-FI DIRECT VULNERABILITY	SURFACE HUB MITIGATION
By spoofing or cloning the wireless name or "SSID" of the target network, an attacker may trick the user into connecting to fake malicious network. By supporting unauthenticated, auto-join Miracast an attacker could capture the intended display materials or attempt to perform network attacks on the connecting device.	While no specific protections against joining a spoofed Surface Hub are in place, this attack is partially mitigated in two ways. First, any potential attack must be physically within Wi-Fi range. Second, this attack is only possible during the very first connection. Subsequent connections use a persistent Wi-Fi Direct group and Windows will remember and prioritize this prior connection during future Hub use. (Note: Spoofing the MAC address, Wi-Fi channel and SSID simultaneously was not considered for this report and may result in inconsistent Wi-Fi behavior.) Overall this weakness is a fundamental problem for any 802.11 wireless network not using Enterprise WPA2 protocols such as EAP-TLS or EAP-PWD, which are not supported in Wi-Fi Direct.

Surface Hub hardening guidelines

Surface Hub is designed to facilitate collaboration and allow users to start or join meetings quickly and efficiently. As such, the default Wi-Fi Direct settings for Surface Hub are optimized for this scenario.

For users who require additional security around the wireless interface, we recommend Surface Hub users enable the WPS-PIN security setting. This disables WPS-PBC mode and offers client authentication, and provides the strongest level of protection by preventing any unauthorized connections to Surface Hub.

If concerns remain around authentication and authorization of a Surface Hub, we recommend users connect the device to a separate network, either Wi-Fi (such as a "guest" Wi-Fi network) or using separate Ethernet network (preferably an entirely different physical network, but a VLAN can also provide some added security). Of course, this approach may preclude connections to internal network resources or services, and may require additional network configurations to regain access.

Also recommended:

- Install regular system updates.
- Update the Miracast settings to disable auto-present mode.

Learn more

- Wi-Fi Direct specifications
- Wireless Protected Setup (WPS) specification

Change history for Surface Hub

5/4/2017 • 1 min to read • Edit Online

This topic lists new and updated topics in the Surface Hub Admin Guide.

May 2017

NEW OR CHANGED TOPIC	DESCRIPTION
Online or hybrid deployment using Skype Hybrid Voice environment	New

February 2017

NEW OR CHANGED TOPIC	DESCRIPTION
Useful downloads for Surface Hub administrators	New

January 2017

NEW OR CHANGED TOPIC	DESCRIPTION
How Surface Hub addresses Wi-Fi Direct security issues	New
On-premises deployment (multiple forests)	New
Connect other devices and display with Surface Hub	Added graphics cards verified to work with 84" Surface Hubs and added information about the lengths of cables.
Online deployment	Updated procedures for adding a device account for your Microsoft Surface Hub when you have a pure, online deployment.

December 2016

NEW OR CHANGED TOPIC	DESCRIPTION
Connect other devices and display with Surface Hub	Added information about Bluetooth accessories.
Manage settings with an MDM provider	Updated example procedures to include screenshots.

November 2016

NEW OR CHANGED TOPIC	DESCRIPTION
Differences between Surface Hub and Windows 10 Enterprise	New

NEW OR CHANGED TOPIC	DESCRIPTION
Connect other devices and display with Surface Hub	Added information for Video Out and a table to help select a display method.
Hybrid deployment	Added instructions for creating accounts for Surface Hub in a Skype for Business hybrid environment.

RELEASE: Windows Anniversary Update for Surface Hub (Windows 10, version 1607)

The topics in this library have been updated for Windows 10, version 1607 (also known as Windows Anniversary Update for Surface Hub). These topics had significant updates for this release:

- Windows Updates (Surface Hub)
- Manage settings with an MDM provider (Surface Hub)
- Monitor your Microsoft Surface Hub
- Create provisioning packages (Surface Hub)
- Install apps on your Microsoft Surface Hub
- Device reset (Surface Hub)

October 2016

NEW OR CHANGED TOPIC	DESCRIPTION
Admin group management (Surface Hub)	Add note about automatic enrollment, and update table.
Password management (Surface Hub)	Updates to content.
Create and test a device account (Surface Hub)	Reorganize and streamline guidance on creating a device account.
Introduction to Surface Hub	Move Surface Hub dependencies table to Prepare your environment for Surface Hub.
Prepare your environment for Surface Hub	Add dependency table and reorganize topic.
Local management for Surface Hub settings	New topic.