

# Server Core App Compatibility Feature on Demand

Article • 03/16/2023

The Server Core App Compatibility Feature on Demand (FOD) is an optional feature package that can be added to Server Core installations of Windows Server installations at any time, beginning with Windows Server 2019.

For more information on other Features on Demand, see [Features On Demand](#).

## Why install the App Compatibility FOD?

The App Compatibility Feature on Demand for Server Core improves app compatibility by including a subset of binaries and packages from the Server with Desktop Experience installation option. This optional package is available on a separate ISO, or from Windows Update, but can only be added to Server Core installations and images.

The two primary values the App Compatibility FOD provides are:

- Increases the compatibility of Server Core for server applications already in market or deployed.
- Assists with providing OS components and increased app compatibility of software tools used in acute troubleshooting and debugging scenarios.

Operating system components that are available as part of the Server Core App Compatibility FOD include:

- Microsoft Management Console (mmc.exe)
- Event Viewer (Eventvwr.msc)
- Performance Monitor (PerfMon.exe)
- Resource Monitor (Resmon.exe)
- Device Manager (Devmgmt.msc)
- File Explorer (Explorer.exe)
- Windows PowerShell (Powershell\_ISE.exe)
- Disk Management (Diskmgmt.msc)

- Failover Cluster Manager (CluAdmin.msc)

#### ⓘ Note

Failover Cluster Manager requires adding the Failover Clustering Windows Server feature first, which can be done by running the following command from an an elevated PowerShell session:

PowerShell

```
Install-WindowsFeature -Name Failover-Clustering -  
IncludeManagementTools
```

Beginning with Windows Server 2022, the following components are also available (when using the same version of the App Compatibility FOD):

- Hyper-V Manager (virtmgmt.msc)
- Task Scheduler (taskschd.msc)

## Installing the App Compatibility Feature on Demand

#### ⓘ Important

- The App Compatibility FOD can only be installed on Server Core. Don't attempt to add the Server Core App Compatibility FOD to the Server with Desktop Experience installation option.
- For servers running Windows Server 2022, ensure you have installed the [2022-01 Cumulative Update Preview for Microsoft server operating system version 21H2 for x64-based Systems \(KB5009608\)](#) <sup>↗</sup> or later cumulative update before you install the App Compatibility FOD. You can verify this by checking that the operating system build number is 20348.502 or greater. Prior to this, if you tried to connect to the server using Remote Desktop Protocol (RDP), you could be presented with a black screen and disconnected.




## Connected to the internet

1. If the server can connect to Windows Update, run the following command from an elevated PowerShell session, then restart Windows Server after the command finishes running:

PowerShell

```
Add-WindowsCapability -Online -Name  
ServerCore.AppCompatibility~~~~0.0.1.0
```

## Disconnected from the internet

1. If the server can't connect to Windows Update, instead download the Windows Server Languages and Optional Features ISO image file, and copy the ISO to a shared folder on your local network:
  - If you have a volume license, you can download the Windows Server Languages and Optional Features ISO image file from the same portal where the operating system ISO image file is obtained: [Volume Licensing Service Center](#) .
  - The Windows Server Languages and Optional Features ISO image file is also available on the [Microsoft Evaluation Center](#)  or on the [Visual Studio portal](#)  for subscribers.

### Note

The Languages and Optional Features ISO image file is new for Windows Server 2022. Previous versions of Windows Server use the Features on Demand (FOD) ISO.

2. Sign in with an administrator account on the Server Core computer that is connected to your local network and that you want to add the App Compatibility FOD to.

## Mount the FOD ISO

1. Use `New-PSDrive` from PowerShell, `net use` from Command Prompt, or some other method, to connect to the location of the FOD ISO. For example, in an elevated PowerShell session run the following command:

PowerShell

```
$credential = Get-Credential

New-PSDrive -Name FODShare -PSProvider FileSystem -Root
"\server\share" -Credential $credential
```

2. Copy the FOD ISO to a local folder of your choosing (the copy operation may take some time). Edit the following variables with your folder location and ISO filename, and run the following commands, for example:

```
PowerShell

$isoFolder = "C:\SetupFiles\WindowsServer\ISOs"
$fodIsoFilename = "FOD_ISO_filename.iso"

New-Item -ItemType Directory -Path $isoFolder
Copy-Item -Path "FODShare:\$fodIsoFilename" -Destination $isoFolder -
Verbose
```

3. Mount the FOD ISO by using the following command:

```
PowerShell

$fodIso = Mount-DiskImage -ImagePath "$isoFolder\$fodIsoFilename"
```

4. Run the following command to get the drive letter that the FOD ISO has been mounted to:

```
PowerShell

$fodDriveLetter = ($fodIso | Get-Volume).DriveLetter
```

5. Run the following command (depending on the operating system version):

For Windows Server 2022:

```
PowerShell

Add-WindowsCapability -Online -Name
ServerCore.AppCompatibility~~~~0.0.1.0 -Source
${fodDriveLetter}:\LanguagesAndOptionalFeatures\ -LimitAccess
```

For previous versions of Windows Server:

```
PowerShell
```

```
Add-WindowsCapability -Online -Name  
ServerCore.AppCompatibility~~~~0.0.1.0 -Source ${fodDriveLetter}:\ -  
LimitAccess
```

6. After the progress bar completes, restart the operating system.

## Optionally add Internet Explorer 11 to Server Core

### ⓘ Note

The Server Core App Compatibility FOD is required for the addition of Internet Explorer 11, but Internet Explorer 11 is not required to add the Server Core App Compatibility FOD.

### ⓘ Note

Starting with Windows Server 2022, although Internet Explorer 11 can be added to Server Core installations of Windows Server, **Microsoft Edge** [|](#) should be used instead. Microsoft Edge has **Internet Explorer mode** ("IE mode") built in, so you can access legacy Internet Explorer-based websites and applications straight from Microsoft Edge. Please see [here](#) for information on the lifecycle policy for Internet Explorer.

1. Sign in as Administrator on the Server Core computer that already has the App Compatibility FOD added and the FOD optional package ISO copied locally.
2. Mount the FOD ISO by using the following command. This step assumes that you've already copied the FOD ISO locally. If not, complete steps 1 and 2 from [Mount the FOD ISO](#). The commands follow on from these two steps. Edit the variables with your folder location and ISO filename, and run the following commands, for example:

PowerShell

```
$isoFolder = "C:\SetupFiles\WindowsServer\ISOs"  
$fodIsoFilename = "FOD_ISO_filename.iso"  
  
$fodIso = Mount-DiskImage -ImagePath "$isoFolder\$fodIsoFilename"
```

3. Run the following command to get the drive letter that the FOD ISO has been mounted to:

PowerShell

```
$fodDriveLetter = ($fodIso | Get-Volume).DriveLetter
```

4. Run the following commands (depending on your operating system version), using the `$packagePath` variable as the path to the Internet Explorer .cab file:

For Windows Server 2022:

PowerShell

```
$packagePath =  
"${fodDriveLetter}:\LanguagesAndOptionalFeatures\Microsoft-Windows-  
InternetExplorer-Optional-Package~31bf3856ad364e35~amd64~~.cab"  
  
Add-WindowsPackage -Online -PackagePath $packagePath
```

For previous versions of Windows Server:

PowerShell

```
$packagePath = "${fodDriveLetter}:\Microsoft-Windows-InternetExplorer-  
Optional-Package~31bf3856ad364e35~amd64~~.cab"  
  
Add-WindowsPackage -Online -PackagePath $packagePath
```

5. After the progress bar completes, restart the operating system.

## Release notes and suggestions

### Important

- Packages installed using FoD won't remain in place after an in-place upgrade to a newer Windows Server version. You will have to install them again after the upgrade.
- Alternatively, you can add FoD packages to your upgrade media. Adding packages to your upgrade media ensures that the new version of any FoD package are present after the upgrade completes. For more info, see the

## Adding capabilities and optional packages to an offline WIM Server Core image section.

- After installation of the App Compatibility FOD and reboot of the server, the command console window frame color will change to a different shade of blue.
- If you choose to also install the Internet Explorer 11 optional package, double-clicking to open locally saved .htm files isn't supported. However, you can **right-click** and choose **Open with Internet Explorer**, or you can open it directly from Internet Explorer **File** -> **Open**.
- To further enhance the app compatibility of Server Core with the App Compatibility FOD, the IIS Management Console has been added to Server Core as an optional component. However, it's necessary to first add the App Compatibility FOD to use the IIS Management Console. IIS Management Console relies on the Microsoft Management Console (mmc.exe), which is only available on Server Core with the addition of the App Compatibility FOD. Use the PowerShell cmdlet [Install-WindowsFeature](#) to add IIS Management Console:

PowerShell

```
Install-WindowsFeature -Name Web-Mgmt-Console
```

- As a general point of guidance, when installing applications on Server Core (with or without these optional packages) it's sometimes necessary to use silent install options and instructions.

## Adding to an offline WIM Server Core image

1. Download both the Languages and Optional Features ISO and the Windows Server ISO image files to a local folder on a Windows computer. You can complete these steps on a Windows desktop PC, it doesn't need to be running Windows Server with the Server Core installation option.
  - If you have a volume license, you can download the Windows Server Languages and Optional Features ISO image file from the same portal where the operating system ISO image file is obtained: [Volume Licensing Service Center](#) [↗](#).
  - The Windows Server Languages and Optional Features ISO image file is also available on the [Microsoft Evaluation Center](#) [↗](#) or on the [Visual Studio portal](#) [↗](#) for subscribers.

### ❗ Note

The Languages and Optional Features ISO image file is new for Windows Server 2022. Previous versions of Windows Server use the Features on Demand (FOD) ISO.

2. Mount both the Languages and Optional Features ISO and the Windows Server ISO by running the following commands in an elevated PowerShell session. Edit the variables with your folder location and ISO filename, and run the following commands, for example::

PowerShell

```
$isoFolder = "C:\SetupFiles\WindowsServer\ISOs"
$fodIsoFilename = "FOD_ISO_filename.iso"
$wsIsoFilename = "Windows_Server_ISO_filename.iso"

$fodIso = Mount-DiskImage -ImagePath "$isoFolder\$fodIsoFilename"
$wsIso = Mount-DiskImage -ImagePath "$isoFolder\$wsIsoFilename"
```

3. Run the following command to get the drive letters that the FOD ISO and Windows Server ISO have been mounted to:

PowerShell

```
$fodDriveLetter = ($fodIso | Get-Volume).DriveLetter
$wsDriveLetter = ($wsIso | Get-Volume).DriveLetter
```

4. Copy the contents of the Windows Server ISO file to a local folder, for example, **C:\SetupFiles\WindowsServer\Files**. The copy operation may take some time:

PowerShell

```
$wsFiles = "C:\SetupFiles\WindowsServer\Files"
New-Item -ItemType Directory -Path $wsFiles

Copy-Item -Path ${wsDriveLetter}:*\* -Destination $wsFiles -Recurse
```

5. Get the image name you want to modify within the install.wim file by using the following command. Add your path to the install.wim file to the `$installWimPath` variable, located inside the **sources** folder of the Windows Server ISO file. Note the names of the images available in this install.wim file from the output.

PowerShell



```
$installWimPath =  
"C:\SetupFiles\WindowsServer\Files\sources\install.wim"  
  
Get-WindowsImage -ImagePath $installWimPath
```

6. Mount the install.wim file in a new folder by using the following command replacing the sample variable values with your own, and reusing the `$installWimPath` variable from the previous command.

- `$wimImageName` - Enter the name of the image you want to mount from the output of the previous command. The example here uses **Windows Server 2022 Datacenter**.
- `$wimMountFolder` - Specify an empty folder to use when accessing the contents of the install.wim file.

PowerShell

```
$wimImageName = "Windows Server 2022 Datacenter"  
$wimMountFolder = "C:\SetupFiles\WindowsServer\WIM"  
  
New-Item -ItemType Directory -Path $wimMountFolder  
Set-ItemProperty -Path $installWimPath -Name IsReadOnly -Value $false  
Mount-WindowsImage -ImagePath $installWimPath -Name $wimImageName -Path  
$wimMountFolder
```

7. Add the capabilities and packages you want to the mounted install.wim image by using the following commands (depending on the version), replacing the sample variable values with your own.

- `$capabilityName` - Specify the name of the capability to install (in this case, the **AppCompatibility** capability).
- `$packagePath` - Specify the path to the package to install (in this case, to the **Internet Explorer** cab file).

For Windows Server 2022:

PowerShell

```
$capabilityName = "ServerCore.AppCompatibility~~~~0.0.1.0"  
$packagePath =  
"${fodDriveLetter}:\LanguagesAndOptionalFeatures\Microsoft-Windows-  
InternetExplorer-Optional-Package~31bf3856ad364e35~amd64~~.cab"  
  
Add-WindowsCapability -Path $wimMountFolder -Name $capabilityName -
```

```
Source "${fodDriveLetter}:\LanguagesAndOptionalFeatures" -LimitAccess
Add-WindowsPackage -Path $wimMountFolder -PackagePath $packagePath
```

For previous versions of Windows Server:

PowerShell

```
$capabilityName = "ServerCore.AppCompatibility~~~~0.0.1.0"
$packagePath = "${fodDriveLetter}:\Microsoft-Windows-InternetExplorer-
Optional-Package~31bf3856ad364e35~amd64~~.cab"

Add-WindowsCapability -Path $wimMountFolder -Name $capabilityName -
Source "${fodDriveLetter}:\\" -LimitAccess
Add-WindowsPackage -Path $wimMountFolder -PackagePath $packagePath
```

8. Dismount and commit changes to the install.wim file by using the following command, which uses the `$wimMountFolder` variable from previous commands:

PowerShell

```
Dismount-WindowsImage -Path $wimMountFolder -Save
```

You can now upgrade your server by running setup.exe from the folder you created for the Windows Server installation files, in this example:

**C:\SetupFiles\WindowsServer\Files.** This folder now contains the Windows Server installation files with the extra capabilities and optional packages included.