Windows Sandbox configuration

Article • 05/25/2023 • Applies to: ✅ Windows 11, ✅ Windows 10

Windows Sandbox supports simple configuration files, which provide a minimal set of customization parameters for Sandbox. This feature can be used with Windows 10 build 18342 or Windows 11. Windows Sandbox configuration files are formatted as XML and are associated with Sandbox via the `.wsb` file extension.

A configuration file enables the user to control the following aspects of Windows Sandbox:

- **vGPU (virtualized GPU)**: Enable or disable the virtualized GPU. If vGPU is disabled, the sandbox will use Windows Advanced Rasterization Platform (WARP).
- **Networking**: Enable or disable network access within the sandbox.
- **Mapped folders**: Share folders from the host with *read* or *write* permissions. Exposing host directories may allow malicious software to affect the system or steal data.
- **Logon command**: A command that's executed when Windows Sandbox starts.
- **Audio input**: Shares the host's microphone input into the sandbox.
- **Video input**: Shares the host's webcam input into the sandbox.
- **Protected client**: Places increased security settings on the RDP session to the sandbox.
- **Printer redirection**: Shares printers from the host into the sandbox.
- **Clipboard redirection**: Shares the host clipboard with the sandbox so that text and files can be pasted back and forth.
- **Memory in MB**: The amount of memory, in megabytes, to assign to the sandbox.

> **Note**
>
> The size of the sandbox window currently isn't configurable.

Creating a configuration file

To create a configuration file:

1. Open a plain text editor or source code editor (for example, Notepad, Visual Studio Code, etc.)

2. Insert the following lines:

   XML

   ```xml
   <Configuration>
   </Configuration>
   ```

3. Add appropriate configuration text between the two lines. For details, see the correct syntax and the examples below.

4. Save the file with the desired name, but make sure its filename extension is `.wsb`. In Notepad, you should enclose the filename and the extension inside double quotation marks, for example, `"My config file.wsb"`.

Using a configuration file

To use a configuration file, double-click it to start Windows Sandbox according to its settings. You can also invoke it via the command line as shown here:

batch

```batch
C:\Temp> MyConfigFile.wsb
```

Keywords, values, and limits

vGPU

Enables or disables GPU sharing.

```
<vGPU>value</vGPU>
```

Supported values:

- *Enable*: Enables vGPU support in the sandbox.
- *Disable*: Disables vGPU support in the sandbox. If this value is set, the sandbox will use software rendering, which may be slower than virtualized GPU.
- *Default* This value is the default value for vGPU support. Currently, this default value denotes that vGPU is disabled.

> **Note**
>
> Enabling virtualized GPU can potentially increase the attack surface of the sandbox.

Networking

Enables or disables networking in the sandbox. You can disable network access to decrease the attack surface exposed by the sandbox.

```
<Networking>value</Networking>
```

Supported values:

- *Enable*: Enables networking in the sandbox.
- *Disable*: Disables networking in the sandbox.
- *Default*: This value is the default value for networking support. This value enables networking by creating a virtual switch on the host and connects the sandbox to it via a virtual NIC.

> **Note**
>
> Enabling networking can expose untrusted applications to the internal network.

Mapped folders

An array of folders, each representing a location on the host machine that will be shared into the sandbox at the specified path. At this time, relative paths aren't supported. If no path is specified, the folder will be mapped to the container user's desktop.

XML

```xml
<MappedFolders>
  <MappedFolder>
    <HostFolder>absolute path to the host folder</HostFolder>
    <SandboxFolder>absolute path to the sandbox folder</SandboxFolder>
    <ReadOnly>value</ReadOnly>
  </MappedFolder>
  <MappedFolder>
    ...
  </MappedFolder>
</MappedFolders>
```

HostFolder: Specifies the folder on the host machine to share into the sandbox. The folder must already exist on the host, or the container will fail to start.

SandboxFolder: Specifies the destination in the sandbox to map the folder to. If the folder doesn't exist, it will be created. If no sandbox folder is specified, the folder will be mapped to the container desktop.

ReadOnly: If *true*, enforces read-only access to the shared folder from within the container. Supported values: *true*/*false*. Defaults to *false*.

> Note
>
> Files and folders mapped in from the host can be compromised by apps in the sandbox or potentially affect the host.

Logon command

Specifies a single command that will be invoked automatically after the sandbox logs on. Apps in the sandbox are run under the container user account. The container user account should be an administrator account.

XML

```xml
<LogonCommand>
  <Command>command to be invoked</Command>
</LogonCommand>
```

Command: A path to an executable or script inside the container that will be executed after signing in.

> Note
>
> Although very simple commands will work (such as launching an executable or script), more complicated scenarios involving multiple steps should be placed into a script file. This script file may be mapped into the container via a shared folder, and then executed via the *LogonCommand* directive.

Audio input

Enables or disables audio input to the sandbox.

```
<AudioInput>value</AudioInput>
```

Supported values:

- *Enable*: Enables audio input in the sandbox. If this value is set, the sandbox will be able to receive audio input from the user. Applications that use a microphone may require this capability.
- *Disable*: Disables audio input in the sandbox. If this value is set, the sandbox can't receive audio input from the user. Applications that use a microphone may not function properly with this setting.
- *Default*: This value is the default value for audio input support. Currently, this default value denotes that audio input is enabled.

> Note
>
> There may be security implications of exposing host audio input to the container.

Video input

Enables or disables video input to the sandbox.

```
<VideoInput>value</VideoInput>
```

Supported values:

- *Enable*: Enables video input in the sandbox.
- *Disable*: Disables video input in the sandbox. Applications that use video input may not function properly in the sandbox.
- *Default*: This value is the default value for video input support. Currently, this default value denotes that video input is disabled. Applications that use video input may not function properly in the sandbox.

> Note
>
> There may be security implications of exposing host video input to the container.

Protected client

When Protected Client mode is enabled, Sandbox adds a new layer of security boundary by running inside an AppContainer Isolation execution environment.

AppContainer Isolation provides Credential, Device, File, Network, Process, and Window isolation.

```
<ProtectedClient>value</ProtectedClient>
```

Supported values:

- *Enable*: Runs Windows sandbox in Protected Client mode. If this value is set, the Sandbox runs in AppContainer Isolation.
- *Disable*: Runs the Sandbox in the standard mode without extra security mitigations.
- *Default*: This value is the default value for Protected Client mode. Currently, this default value denotes that the sandbox doesn't run in Protected Client mode.

> ⓘ **Note**
>
> This setting may restrict the user's ability to copy/paste files in and out of the sandbox.

Printer redirection

Enables or disables printer sharing from the host into the sandbox.

```
<PrinterRedirection>value</PrinterRedirection>
```

Supported values:

- *Enable*: Enables sharing of host printers into the sandbox.
- *Disable*: Disables printer redirection in the sandbox. If this value is set, the sandbox can't view printers from the host.
- *Default*: This value is the default value for printer redirection support. Currently, this default value denotes that printer redirection is disabled.

Clipboard redirection

Enables or disables sharing of the host clipboard with the sandbox.

```
<ClipboardRedirection>value</ClipboardRedirection>
```

Supported values:

- *Enable*: Enables sharing of the host clipboard with the sandbox.
- *Disable*: Disables clipboard redirection in the sandbox. If this value is set, copy/paste in and out of the sandbox will be restricted.
- *Default*: This value is the default value for clipboard redirection. Currently, copy/paste between the host and sandbox are permitted under *Default*.

Memory in MB

Specifies the amount of memory that the sandbox can use in megabytes (MB).

```
<MemoryInMB>value</MemoryInMB>
```

If the memory value specified is insufficient to boot a sandbox, it will be automatically increased to the required minimum amount.

Example 1

The following config file can be used to easily test the downloaded files inside the sandbox. To achieve this testing, networking and vGPU are disabled, and the sandbox is allowed read-only access to the shared downloads folder. For convenience, the logon command opens the downloads folder inside the sandbox when it's started.

Downloads.wsb

XML

```xml
<Configuration>
  <VGpu>Disable</VGpu>
  <Networking>Disable</Networking>
  <MappedFolders>
    <MappedFolder>
      <HostFolder>C:\Users\Public\Downloads</HostFolder>
      <SandboxFolder>C:\Users\WDAGUtilityAccount\Downloads</SandboxFolder>
      <ReadOnly>true</ReadOnly>
    </MappedFolder>
  </MappedFolders>
  <LogonCommand>
    <Command>explorer.exe C:\users\WDAGUtilityAccount\Downloads</Command>
  </LogonCommand>
</Configuration>
```

Example 2

The following config file installs Visual Studio Code in the sandbox, which requires a slightly more complicated LogonCommand setup.

Two folders are mapped into the sandbox; the first (SandboxScripts) contains VSCodeInstall.cmd, which will install and run Visual Studio Code. The second folder (CodingProjects) is assumed to contain project files that the developer wants to modify using Visual Studio Code.

With the Visual Studio Code installer script already mapped into the sandbox, the LogonCommand can reference it.

VSCodeInstall.cmd

Download vscode to `downloads` folder and run from `downloads` folder.

batch

```batch
REM Download Visual Studio Code
curl -L "https://update.code.visualstudio.com/latest/win32-x64-user/stable" --output C:\users
\WDAGUtilityAccount\Downloads\vscode.exe

REM Install and run Visual Studio Code
C:\users\WDAGUtilityAccount\Downloads\vscode.exe /verysilent /suppressmsgboxes
```

VSCode.wsb

XML

```xml
<Configuration>
  <MappedFolders>
    <MappedFolder>
      <HostFolder>C:\SandboxScripts</HostFolder>
      <SandboxFolder>C:\Users\WDAGUtilityAccount\Downloads\sandbox</SandboxFolder>
      <ReadOnly>true</ReadOnly>
    </MappedFolder>
    <MappedFolder>
      <HostFolder>C:\CodingProjects</HostFolder>
      <SandboxFolder>C:\Users\WDAGUtilityAccount\Documents\Projects</SandboxFolder>
      <ReadOnly>false</ReadOnly>
    </MappedFolder>
  </MappedFolders>
  <LogonCommand>
    <Command>C:\Users\WDAGUtilityAccount\Downloads\sandbox\VSCodeInstall.cmd</Command>
  </LogonCommand>
</Configuration>
```

Example 3

The following config file runs a PowerShell script as a logon command to swap the primary mouse button for left-handed users.

`C:\sandbox` folder on the host is mapped to the `C:\sandbox` folder in the sandbox, so the `SwapMouse.ps1` script can be referenced in the sandbox configuration file.

SwapMouse.ps1

Create a powershell script using the following code, and save it in the `C:\sandbox` directory as `SwapMouse.ps1`.

PowerShell

```
[Reflection.Assembly]::LoadWithPartialName("System.Windows.Forms") | Out-Null

$SwapButtons = Add-Type -MemberDefinition @'
[DllImport("user32.dll")]
public static extern bool SwapMouseButton(bool swap);
'@ -Name "NativeMethods" -Namespace "PInvoke" -PassThru

$SwapButtons::SwapMouseButton(!([System.Windows.Forms.SystemInformation]::MouseButtonsSwapped))
```

SwapMouse.wsb

XML

```
<Configuration>
  <MappedFolders>
    <MappedFolder>
      <HostFolder>C:\sandbox</HostFolder>
      <SandboxFolder>C:\sandbox</SandboxFolder>
      <ReadOnly>True</ReadOnly>
    </MappedFolder>
  </MappedFolders>
  <LogonCommand>
    <Command>powershell.exe -ExecutionPolicy Bypass -File C:\sandbox\SwapMouse.ps1</Command>
  </LogonCommand>
</Configuration>
```