Learn

Discover ⌄ | Product documentation ⌄ | Development languages ⌄ | Topics ⌄

Sign in

Visual Studio IDE   Edit   Build   Debug   Test   Deploy   Common tasks ⌄   Troubleshooting   Resources ⌄   **Download Visual Studio**

Learn / Visual Studio / Deployment /

C# ⌄   ⊕   ✏️   ⋮

Configure the ClickOnce trust prompt behavior

Article • 10/23/2024 • 10 contributors

🖒 Feedback

In this article

Enable the ClickOnce trust prompt

Restrict the ClickOnce trust prompt

Disable the ClickOnce trust prompt

Related content

You can configure the ClickOnce trust prompt to control whether end users are given the option of installing ClickOnce applications, such as Windows Forms applications, Windows Presentation Foundation applications, console applications, WPF browser applications, and Office solutions. You configure the trust prompt by setting registry keys on each end user's computer.

The following table shows the configuration options that can be applied to each of the five zones (Internet, UntrustedSites, MyComputer, LocalIntranet, and TrustedSites).

⛶ Expand table

Option	Registry setting value	Description
Enable the trust prompt.	`Enabled`	The ClickOnce trust prompt is displayed so that end users can grant trust to ClickOnce applications.
Restrict the trust prompt.	`AuthenticodeRequired`	The ClickOnce trust prompt is only displayed if ClickOnce applications are signed with a certificate that identifies the publisher. Otherwise, the ClickOnce application won't be installed.
Disable the trust prompt.	`Disabled`	The ClickOnce trust prompt isn't displayed. Only ClickOnce applications that are signed with an explicitly trusted certificate will be installed.

The following table shows the default behavior for each zone. The Applications column refers to Windows Forms applications, Windows Presentation Foundation applications, WPF browser applications, and console applications.

⛶ Expand table

Zone	Applications	Office solutions
`MyComputer`	`Enabled`	`Enabled`
`LocalIntranet`	`Enabled`	`Enabled`
`TrustedSites`	`Enabled`	`Enabled`
`Internet`	`Enabled`	`AuthenticodeRequired`
`UntrustedSites`	`Disabled`	`Disabled`

You can override these settings by enabling, restricting, or disabling the ClickOnce trust prompt.

Enable the ClickOnce trust prompt

Enable the trust prompt for a zone when you want end users to be presented with the option of installing and running any ClickOnce application that comes from that zone.

To enable the ClickOnce trust prompt by using the registry editor

1. Open the registry editor:

   a. Click **Start**, and then click **Run**.

   b. In the **Open** box, type `regedit`, and then click **OK**.

2. Find the following registry key:

   \HKEY_LOCAL_MACHINE\SOFTWARE\MICROSOFT\.NETFramework\Security\TrustManager\PromptingLevel

   If the key doesn't exist, create it.

3. Add the following subkeys as **String Value**, if they don't already exist, with the associated values shown in the following table.

⌞⌝ **Expand table**

String Value subkey	Value
`Internet`	`Enabled`
`UntrustedSites`	`Disabled`
`MyComputer`	`Enabled`
`LocalIntranet`	`Enabled`
`TrustedSites`	`Enabled`

For Office solutions, `Internet` has the default value `AuthenticodeRequired` and `UntrustedSites` has the value `Disabled`. For all others, `Internet` has the default value `Enabled`.

To enable the ClickOnce trust prompt programmatically

1. Create a Visual Basic or Visual C# console application in Visual Studio.

2. Open the *Program.vb* or *Program.cs* file for editing and add the following code.

C#	VB

   C#   Copy

   ```csharp
   Microsoft.Win32.RegistryKey key;
   key = Microsoft.Win32.Registry.LocalMachine.CreateSubKey("SOFTWARE\\MIC
   key.SetValue("MyComputer", "Enabled");
   key.SetValue("LocalIntranet", "Enabled");
   key.SetValue("Internet", "AuthenticodeRequired");
   key.SetValue("TrustedSites", "Enabled");
   key.SetValue("UntrustedSites", "Disabled");
   key.Close();
   ```

3. Build and run the application.

Restrict the ClickOnce trust prompt

Restrict the trust prompt so that solutions must be signed with Authenticode certificates that have known identity before users are prompted for a trust decision.
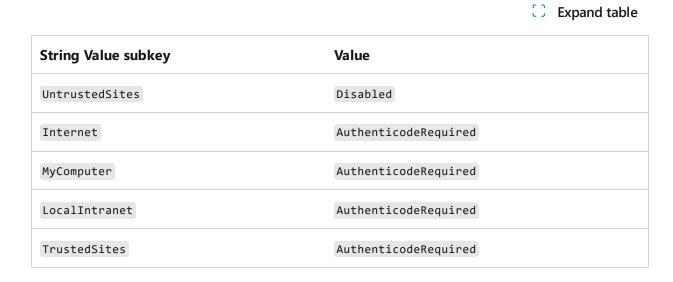
To restrict the ClickOnce trust prompt by using the registry editor

1. Open the registry editor:

   a. Click **Start**, and then click **Run**.

   b. In the **Open** box, type `regedit`, and then click **OK**.

2. Find the following registry key:

   \HKEY_LOCAL_MACHINE\SOFTWARE\MICROSOFT\.NETFramework\Security\TrustManager\PromptingLevel

   If the key doesn't exist, create it.

3. Add the following subkeys as **String Value**, if they don't already exist, with the associated values shown in the following table.

⤢ Expand table

String Value subkey	Value
`UntrustedSites`	`Disabled`
`Internet`	`AuthenticodeRequired`
`MyComputer`	`AuthenticodeRequired`
`LocalIntranet`	`AuthenticodeRequired`
`TrustedSites`	`AuthenticodeRequired`

To restrict the ClickOnce trust prompt programmatically

1. Create a Visual Basic or Visual C# console application in Visual Studio.

2. Open the *Program.vb* or *Program.cs* file for editing and add the following code.

C# | VB

```csharp
C#                                                              Copy

Microsoft.Win32.RegistryKey key;
key = Microsoft.Win32.Registry.LocalMachine.CreateSubKey("SOFTWARE\\MIC
key.SetValue("MyComputer", "AuthenticodeRequired");
key.SetValue("LocalIntranet", "AuthenticodeRequired");
key.SetValue("Internet", "AuthenticodeRequired");
key.SetValue("TrustedSites", "AuthenticodeRequired");
key.SetValue("UntrustedSites", "Disabled");
key.Close();
```

3. Build and run the application.

Disable the ClickOnce trust prompt

You can disable the trust prompt so that end users aren't given the option to install solutions that aren't already trusted in their security policy.
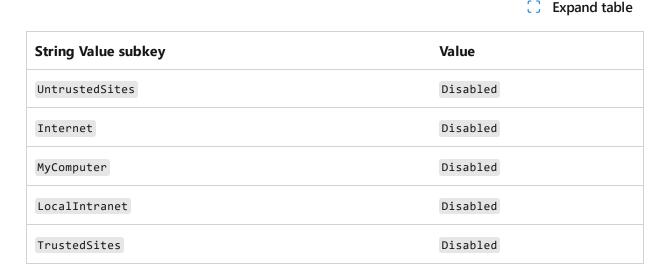
To disable the ClickOnce trust prompt by using the registry editor

1. Open the registry editor:

   a. Click **Start**, and then click **Run**.

   b. In the **Open** box, type `regedit`, and then click **OK**.

2. Find the following registry key:

   \HKEY_LOCAL_MACHINE\SOFTWARE\MICROSOFT\.NETFramework\Security\TrustManager\PromptingLevel

   If the key doesn't exist, create it.

3. Add the following subkeys as **String Value**, if they don't already exist, with the associated values shown in the following table.

⌞⌝ **Expand table**

String Value subkey	Value
`UntrustedSites`	`Disabled`
`Internet`	`Disabled`
`MyComputer`	`Disabled`
`LocalIntranet`	`Disabled`
`TrustedSites`	`Disabled`

To disable the ClickOnce trust prompt programmatically

1. Create a Visual Basic or Visual C# console application in Visual Studio.

2. Open the *Program.vb* or *Program.cs* file for editing and add the following code.

C# | VB

```
C#                                                                    Copy

Microsoft.Win32.RegistryKey key;
key = Microsoft.Win32.Registry.LocalMachine.CreateSubKey("SOFTWARE\\MIC
key.SetValue("MyComputer", "Disabled");
key.SetValue("LocalIntranet", "Disabled");
key.SetValue("Internet", "Disabled");
key.SetValue("TrustedSites", "Disabled");
key.SetValue("UntrustedSites", "Disabled");
key.Close();
```

3. Build and run the application.

Related content

- Secure ClickOnce applications
- Code access security for ClickOnce applications
- ClickOnce and Authenticode

- Trusted application deployment overview
- Enable and configure ClickOnce security settings
- Set a security zone for a ClickOnce application
- Set custom permissions for a ClickOnce application
- Debug a ClickOnce application with restricted permissions
- Add a trusted publisher to a client computer for ClickOnce applications
- Re-sign application and deployment manifests

Feedback

Was this page helpful? 👍 Yes   👎 No

Provide product feedback ⧉ | Ask the community ⧉

Additional resources

🎓 Training

Module

Configure User Account Control - Training

This module introduces how User Account Control works and how you can use UAC-related desktop features to reduce security risks.

🌐 English (United States)   ☑☒ Your Privacy Choices   ☼ Theme ⌄

Manage cookies   Previous Versions   Blog ⧉   Contribute   Privacy ⧉   Terms of Use   Trademarks ⧉   © Microsoft 2024