

# Specifying File Handlers for File Name Extensions - Visual Studio 2015

By mijacobs

Archived: 2026-04-06 00:30:50 UTC

## Note

This article applies to Visual Studio 2015. If you're looking for the latest Visual Studio documentation, see [Visual Studio documentation](#). We recommend upgrading to the latest version of Visual Studio. [Download it here](#)

There are a number of ways to determine the application that handles a file which has a particular file extension. The `OpenWithList` and `OpenWithProgids` verbs are two ways to specify file handlers under the registry entry for the file extension.

When you right-click a file in Windows Explorer, you see the **Open** command. If more than one product is associated with an extension, you see an **Open With** submenu.

You can register different applications to open an extension by setting the `OpenWithList` key for the file extension in `HKEY_CLASSES_ROOT`. The applications listed under this key for a file extension appear under the **Recommended Programs** heading in the **Open With** dialog box. The following example shows the applications registered to open the `.vcproj` file extension.

```
HKEY_CLASSES_ROOT\  
  .vcproj\  
    (default)="VisualStudio.vcproj.14.0"  
    OpenWithList\  
      devenv.exe
```

## Note

The keys specifying applications are from the list under `HKEY_CLASSES_ROOT\Applications`.

By adding an `OpenWithList` key, you declare that your application supports a file extension even if another application takes ownership of the extension. This could be a future version of your application or another application.

Programmatic identifiers (ProgIDs) are friendly versions of ClassIDs that identify a version of an application or COM object. Every co-creatable object should have its own ProgID. For example, `VisualStudio.DTE.7.1` starts Visual Studio .NET 2003 while `VisualStudio.DTE.10.0` starts Visual Studio. As the owner of a project type or project item type, you must create a version-specific ProgID for your file extension. These ProgIDs may be

redundant in that more than one ProgID may start the same application. For more information, see [Registering Verbs for File Name Extensions](#).

Use the following naming convention for versioned file ProgIDs to avoid duplication with registration from other vendors:

File extension	Versioned ProgID
.extension	ProductName.extension.versionMajor.versionMinor

You can register different applications that are able to open a particular file extension by adding versioned ProgIDs as values to the HKEY\_CLASSES\_ROOT\*<extension>*\OpenWithProgids key. This registry key contains a list of alternate ProgIDs associated with the file extension. The applications associated with the listed ProgIDs appear in the **Open With***Product Name* submenu. If the same application is specified in both the `OpenWithList` and `OpenWithProgids` keys, the operating system merges the duplicates.

Note

The `OpenWithProgids` key is only supported in Windows XP. Because other operating systems ignore this key, do not use it as the only registration for file handlers. Use this key to provide a better user experience in Windows XP.

Add the desired ProgIDs as values of the type REG\_NONE. The following code provides an example of registering ProgIDs for a file extension (.ext).

```
HKEY_CLASSES_ROOT\
  .ext\
    (default)="MyProduct.ext.14.0"
    OpenWithProgids
      progid      REG_NONE (zero-length binary value)
      otherprogid REG_NONE (zero-length binary value)
```

The ProgID specified as the default value for the file extension is the default file handler. If you modify the ProgID for a file extension that shipped with a previous version of Visual Studio or that can be taken over by other applications, then you must register the `OpenWithProgids` key for your file extension and specify the new ProgID in the list along with the old ProgIDs you support. For example:

```
HKEY_CLASSES_ROOT\
  .vcproj\
    (default)="VisualStudio.vcproj.14.0"
    OpenWithProgids
      vcprojfile      //old progid
      VisualStudio.vcproj.12.0 //old progid
      VisualStudio.vcproj.14.0 //new progid
```

If the old ProgID has verbs associated with it, then these verbs will also appear under **Open With** *Product Name* in the shortcut menu.

[About File Name Extensions](#)

[Registering Verbs for File Name Extensions](#)

---

Source: <http://msdn.microsoft.com/en-us/library/bb166549.aspx>