11,545,482 members (62,317 online)

1 dzzie 281 Sign out

Article

Browse Code

Stats

Revisions

Alternatives (1)

Comments (75)

home articles quick answers discussions features community help

Search for articles, questions, tips

Articles » Platforms, Frameworks & Libraries » COM / COM+ » Tools

Add your own alternative version

regsvr42: Generate SxS Manifest Files from Native DLLs for Registration-Free COM

Cristian Adam, 19 Aug 2008 Zlib 147.3K 6.1K 83 Rate:

★★★★★ 4.94 (33 votes)

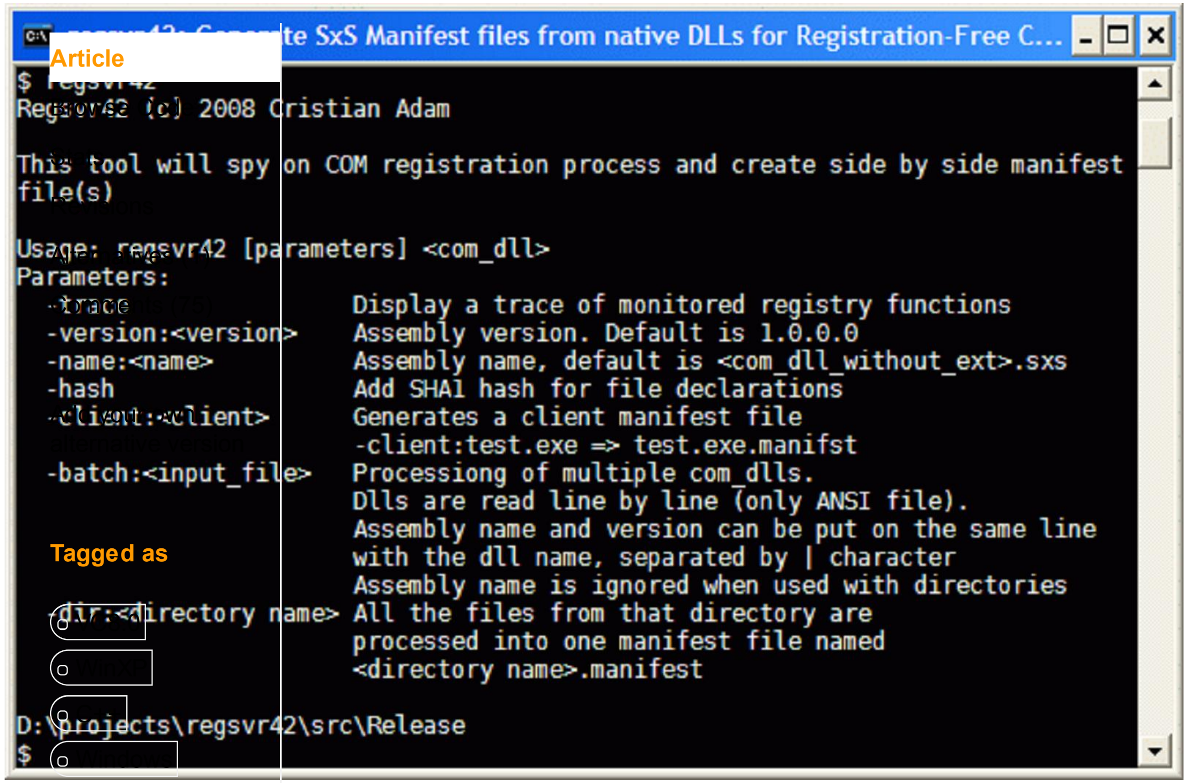
This tool will spy on COM registration process and create side by side (SxS) manifest file(s)

Is your email address OK? You are signed up for our newsletters but your email address is either unconfirmed, or has not been reconfirmed in a long time. Please click **here** to have a confirmation email sent so we can confirm your email address and start sending you newsletters again. Alternatively, you can **update your subscriptions**.

Download test application - 71.73 KB

Download source - 29.46 KB

Download tool - 99.66 KB



regsvr42: Generate SxS Manifest files from native DLLs for Registration-Free C...

```
$ regsvr42
Regsvr42 (c) 2008 Cristian Adam

This tool will spy on COM registration process and create side by side manifest
file(s)

Usage: regsvr42 [parameters] <com_dll>
Parameters:
  -trace                  Display a trace of monitored registry functions
  -version:<version>      Assembly version. Default is 1.0.0.0
  -name:<name>            Assembly name, default is <com_dll_without_ext>.sxs
  -hash                   Add SHA1 hash for file declarations
  -client:<client>        Generates a client manifest file
                          -client:test.exe => test.exe.manifst
  -batch:<input_file>     Processiong of multiple com_dlls.
                          Dlls are read line by line (only ANSI file).
                          Assembly name and version can be put on the same line
                          with the dll name, separated by | character
                          Assembly name is ignored when used with directories
  -dir:<directory name>   All the files from that directory are
                          processed into one manifest file named
                          <directory name>.manifest

D:\projects\regsvr42\src\Release
$
```

Introduction

Windows XP introduced a new COM activation model called Registration-Free COM activation. The Registration-Free COM activation is a registry replacement for COM components. The registry information will reside in a DotManifest file that can be stored in the same folder as the application itself.

This means that you don't have to have information into registry, information which is normally stored into *HKEY_LOCAL_MACHINE*, thus enabling regular user accounts to use COM DLLs without registering them into the system.

The problem with these DotManifest files is that they are pretty hard to write by hand. For managed assemblies there is GenMan32 - A tool to generate Sxs manifest for managed assembly for Registration Free COM/.NET Interop, but there is no tool for native DLLs, this is why I've decided to write such a tool.

Article

Tagged as

- Dev
- COM
- Intermediate
- VC9.0

Related Articles

regsvr42: Generate SxS Manifest Files from Native DLLs for Registration-Free COM

Generating IDL and Manifest Files with

Background

The following MSDN article will give the proper background information: Registration-Free Activation of COM Components: A Walkthrough by Steve White and Leslie Muller. Also of great help is the MSDN Side-by-side Assemblies Reference.

A DotManifest file is a XML file — you probably have heard about these in conjunction with Common Controls version 6, or with Windows Vista elevation privileges and not to mention the (in)famous `Microsoft.VC[8|9]0.CRT.manifest` and `Microsoft.VC[8|9]0.MFC.manifest` files. These days every application has embedded a DotManifest file inside by the Manifest Tool as a resource type 24 (RT_MANIFEST). A typical DotManifest file looks like:

Hide Shrink ▲ Copy Code

```
<?xml version='1.0' encoding='UTF-8' standalone='yes'?>
<assembly xmlns='urn:schemas-microsoft-com:asm.v1' manifestVersion='1.0'>
  <trustInfo xmlns="urn:schemas-microsoft-com:asm.v3">
    <security>
      <requestedPrivileges>
        <requestedExecutionLevel level='asInvoker' uiAccess='false' />
      </requestedPrivileges>
    </security>

  </trustInfo>
  <dependency>
    <dependentAssembly>
      <assemblyIdentity type='win32' name='Microsoft.VC90.CRT' version='9.0.21022.8'
          processorArchitecture='x86' publicKeyToken='1fc8b3b9a1e18e3b' />
    </dependentAssembly>
  </dependency>
  <dependency>
    <dependentAssembly>
      <assemblyIdentity type='win32' name='Microsoft.VC90.MFC' version='9.0.21022.8'
          processorArchitecture='x86' publicKeyToken='1fc8b3b9a1e18e3b' />

    </dependentAssembly>
  </dependency>
  <dependency>
    <dependentAssembly>
      <assemblyIdentity type='win32' name='Microsoft.Windows.Common-Controls'
          version='6.0.0.0' processorArchitecture='x86' publicKeyToken='6595b64144ccf1df'
          language="'*'" />
    </dependentAssembly>
  </dependency>
```

Article

Browse Code

Stats

Revisions

Alternatives (1)

Comments (75)

Add your own alternative version

Tagged as

VC8.0

WinXP

C++

Windows

Dev

COM

Intermediate

VC9.0

Related Articles

regsvr42: Generate SxS Manifest Files from Native DLLs for Registration-Free COM

Generating IDL and Manifest Files with

```
</assembly>
```

The Spying

In order to get the the information about the COM registration process I had to "intercept" the following registry access functions:

- RegCreateKeyA
- RegCreateKeyW
- RegCreateKeyExA
- RegCreateKeyExW
- RegSetValueA
- RegSetValueW
- RegSetValueExA
- RegSetValueExW
- RegOpenKeyA
- RegOpenKeyW
- RegOpenKeyExA
- RegOpenKeyExW
- RegCloseKey

The interception was done by using the class CAPIHook presented in *Chapter 22: DLL Injection and API Hooking* of the book **Windows via C/C++, Fifth Edition** written *by Jeffrey Richter and Christophe Nasarre*, Microsoft Press (c) 2008. One can use the Detours library from Microsoft Research instead of CAPIHook. I've decided for the latter because of simpler deployment (just the executable, no signature DLL).

Setting up a CAPIHook is being done like this

Hide Copy Code

```
std::auto_ptr<CAPIHook> Interceptor::m_RegCreateKeyA;

...

    if (!m_RegCreateKeyA.get())
    {
        m_RegCreateKeyA.reset(new CAPIHook("Advapi32.dll", "RegCreateKeyA",
            (PROC)RegCreateKeyA));
    }
```

The upper code will redirect RegCreateKeyA through Interceptor::RegCreateKeyA, the original function can

Article
Browse Code
Stats
Revisions
Alternatives (1)
Comments (75)

Add your own alternative version

Tagged as

VC8.0
WinXP
C++
Windows
Dev
COM
Intermediate
VC9.0

Related Articles

regsvr42: Generate SxS Manifest Files from Native DLLs for Registration-Free COM

Generating DLL and Manifest Files with

be reached through m_RegCreateKeyA. Interceptor::RegCreateKeyA, like this:

Hide Copy Code

```
LONG WINAPI Interceptor::RegCreateKeyA(HKEY hKey, LPCSTR lpSubKey, PHKEY phkResult)
{
    if (m_doTrace)
    {
        PrintKeyStats(hKey, lpSubKey, __FUNCTIONW__);
    }

    // just to have it initialized
    LONG result = ERROR_ARENA_TRASHED;
    try
    {
        result = ((PFNREGCREATEKEYA)(PROC)*m_RegCreateKeyA)(hKey, lpSubKey, phkResult);
    }
    catch (...)
    {
    }

    if (result == ERROR_SUCCESS)
    {
        InsertSubkeyIntoUserKeyMap(hKey, *phkResult, lpSubKey, __FUNCTIONW__);
    }

    return result;
}
```

After all the information was acquired then the information is processed and written into DotManifest files. The manifest files are encoded as UTF-8. I would like to point fprintf's ccs=<encoding> parameter which was used to save the manifest files as UTF-8.

Hide Copy Code

```
void ManifestWriter::WriteToFile(const std::wstring& outputManifestFile)
{
    FILE* file = _wfopen(outputManifestFile.c_str(), L"w, ccs=utf-8");
    if (file)
    {
        fwrite(&m_data.str().begin(), 1, m_data.str().size() * 2, file);
        fclose(file);
    }
}
```

In order to get all the information possible, first DllUnregisterServer function is called, then DllRegisterServer and at the end again DllUnregisterServer. If you use **regsvr42** on a system where

Article

Browse Code

Stats

Revisions

Alternatives (1)

Comments (75)

Add your own alternative version

Tagged as

VC8.0

WinXP

C++

Windows

Dev

COM

Intermediate

VC9.0

Related Articles

regsvr42: Generate SxS Manifest Files from Native DLLs for Registration-Free COM

Generating IDL and Manifest Files with

you already have the COM DLL registered and still want to use it normally don't forget to run **regsvr32** for that CO

Tool Usage

The basic usage is:

regsvr42 com.dll

which will generate a file named *com.sxs.manifest*. You can find out what interfaces and coclasses the COM DLL exports.

When used with a client application the usage is:

regsvr42 -client:client.exe com.dll

which will generate besides *com.sxs.manifest* another manifest file named *client.exe.manifest*. If *client.exe* already has a manifest file embedded, the contents of that manifest file are preserved into *client.exe.manifest* alongside with the reference to *com.sxs* assembly.

If you have more than one COM DLLs you want to use can use the tool in batch mode like:

regsvr42 -client:client.exe -batch:file_containing_com_dll_file_names

You can put all the COM DLLs inside one directory and there will be just one manifest file inside the directory named *directory_name.manifest*

regsvr42 -client:client.exe -dir:directory_with_com_dlls

If you have more than one directories with COM DLLs you can use the *-batch* function with all the names of the directories written in the batch file.

regsvr42 -client:client.exe -batch:file_containing_directory_names

DirectShow Filters

DirectShow filters are COM DLLs. They come in three flavours: source filters, transform filters and render filters.

The source filters add extra information into registry to support DirectShow's intelligent connect, this extra information cannot be stored into a DotManifest file. If your application relays on Intelligent Connect for a source

Article

Browse Code

Stats

Revisions

Alternatives (1)

Comments (75)

Add your own alternative version

Tagged as

VC8.0

WinXP

C++

Windows

Dev

COM

Intermediate

VC9.0

Related Articles

regsvr42: Generate SxS Manifest Files from Native DLLs for Registration-Free COM

Generating IDL and Manifest Files with

filter, it will not work correctly.

There is one solution for this problem. For example instead of using `IGraphBuilder::RenderFile` one should:

- Add the source filter in to the DirectShow graph
- Acquire the `IFileSourceFilter` interface and call the `Load` method
- Acquire the output pin of the source filter and call `IPin::Render`

Test Application

I have made a test application which uses a source filter as described above. The test application will try to play an aacPlus radio station by the use of Orban/Coding Technologies AAC/aacPlus Player Plugin

To test the application you need to have the Orban Plugin installed and the just run *playradio.exe*.

To test the Registration-Free COM mechanism first you have to unregister the Orban Plugin like:

regsvr32 /u "%ProgramFiles%\Orban\AAC-aacPlus Plugin\aacpParser.dll"

After that check to see that *playradio.exe* displays *Class not registered. ErrorCode: 0x80040154*.

Then run *make_manifest.cmd*, a convenience batch file, which does the DotManifest creation like *regsvr42 -client:playradio.exe "%ProgramFiles%\Orban\AAC-aacPlus Plugin\aacpParser.dll"*. Two DotManifest files will be created:

aacpParser.sxs.manifest

Hide Copy Code

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<assembly xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersion="1.0">

<assemblyIdentity
    type="win32"
    name="aacpParser.sxs"
    version="1.0.0.0" />

<file name="C:\\Program Files\\Orban\\AAC-aacPlus Plugin\\aacpParser.dll">

    <comClass
        description="ORBAN-CT AAC/aacPlus Stream Parser"
        clsid="{301F7BDA-B1F8-4453-82B2-0B9187DF3F3F}"
        threadingModel="Both" />
```

Article
Browse Code
Stats
Revisions
Alternatives (1)
Comments (75)

Add your own alternative version

Tagged as

VC8.0
WinXP
C++
Windows
Dev
COM
Intermediate
VC9.0

Related Articles

regsvr42: Generate SxS Manifest Files from Native DLLs for Registration-Free COM

Generating IDL and Manifest Files with

```
              description="ORBAN-CT AAC/aacPlus Stream Parser About Page"
              clsid="{CA920EED-F427-41B8-838F-33FCF47D5306}"
              threadingModel="Both" />
</file>
</assembly>
```

playradio.exe.manifest

Hide Copy Code

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<assembly xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersion="1.0">
  <trustInfo xmlns="urn:schemas-microsoft-com:asm.v3">
    <security>
      <requestedPrivileges>
        <requestedExecutionLevel level="asInvoker" uiAccess="false">
        </requestedExecutionLevel>

      </requestedPrivileges>
    </security>
  </trustInfo>
  <dependency>
        <dependentAssembly>
            <assemblyIdentity
                type="win32"
                name="aacpParser.sxs"
                version="1.0.0.0" />
        </dependentAssembly>
  </dependency>
</assembly>
```

File Hash

By using the command *-hash* the *file* sections will contain a SHA1 hash. To compute the hash I have used CodeProject's CSHA1 by Dominik Reichl. I have verified the validity of the hash by using OpenSSL like this *openssl dgst -sha1 <com.dll>*.

The problem is that the Manifest Tool *mt.exe* doesn't think that the hash is valid! The command used was *mt -manifest aacpParser.sxs.manifest -validate_file_hashes*

Article
Browse Code
Stats
Revisions
Alternatives (1)
Comments (75)

Add your own alternative version

Tagged as

VC8.0
WinXP
C++
Windows
Dev
COM
Intermediate
VC9.0

Related Articles

regsvr42: Generate SxS Manifest Files from Native DLLs for Registration-Free COM

Generating IDL and Manifest Files with

The solution to this problem is to use the Manifest Tool to update the file hash, which renders the *-hash* function use [illegible] *manifest aacpParser.sxs.manifest -hashupdate* If anybody knows how to adapt the `CSHA1` to produce valid Manifest Tool SHA1 hashes please let me know.

Debugging

In case you encounter the following MessageBox:

This application has failed to start because the application configuration is incorrect. Reinstalling the application may fix this problem.

The *Event Viewer - System* category contains information about the loading errors of side by side assemblies.

History

2008-08-17 Initial release.

License

This article, along with any associated source code and files, is licensed under The zlib/libpng License

Share

About the Author

Cristian Adam

Software Developer
Germany

Article

Browse Code

Stats

Revisions

Alternatives (1)

Comments (75)

Add your own alternative version

Tagged as

VC8.0

WinXP

C++

Windows

Dev

COM

Intermediate

VC9.0

EMAIL TWITTER

Related Articles

regsvr42: Generate SxS Manifest Files from Native DLLs for Registration-Free COM

Generating IDL and Manifest Files with

Software engineer by day, Xiph.Org contributor by night.

Article

Browse Code

Stats

Revisions

Alternatives (1)

Comments (75)

Add your own alternative version

Tagged as

VC8.0

WinXP

C++

Windows

Dev

COM

Intermediate

VC9.0

Comments and Discussions

Add a Comment or Question ?

Search Comments Go

☑ Profile popups Spacing Tight Noise Low Layout Normal Per page 50 Update

First Prev Next

Manifests for OPOS CCO's new	Sinspin	19-Mar-14 23:55
Class not registered new	hackerspk	14-Jan-14 17:12
Client manifest creation fails if client.exe is smaller than 10kB new	Member 8654360	5-Nov-12 5:13
My vote of 5 new	Rotted Frog	17-Sep-12 8:35
Problem on Win7 x64 new	tzleo	28-Jun-12 23:11
Re: Problem on Win7 x64 new	gxdata	10-Oct-12 7:44
My vote of 5 new	bartolo	18-Jun-12 2:28
Doesn't work for .Net COM DLLs new	Member 8871361	23-May-12 7:15

Related Articles

regsvr42: Generate SxS Manifest Files from Native DLLs for Registration-Free COM

Generating IDL and Manifest Files with

SHA1 Hash new	Member 838012	26-Apr-12 14:20
new	Cristian Adam	18-Jun-12 0:48
Re: SHA1 Hash new	tzleo	29-Jun-12 1:15
My vote of 5 new	herves	17-Apr-12 6:15
Multiple COMs in one single manifest new	gbessis	29-Feb-12 5:21
My vote of 5 new	manoj kumar choubey	28-Feb-12 20:21
Trick to get Windows 7 to recognise external manifests. new	Member 1708957	8-Sep-11 22:44
Re: Trick to get Windows 7 to recognise external manifests. new	herves	17-Apr-12 10:20
'Could not find DllRegisterServer function!' error new	sprezzatura	31-Aug-11 11:52
Love it, got it working, but does MS give us the equivalent of new ActiveXObject('some.programID') for registration free com new	Member 1003861	1-Jun-11 16:42
Re: Love it, got it working, but does MS give us the equivalent of new ActiveXObject('some.programID') for registration free com new	Cristian Adam	2-Jun-11 3:20
MANY THANKS ! new	Member 7688086	21-Feb-11 6:46
Re: MANY THANKS ! new	Cristian Adam	21-Feb-11 7:01
Genius! new	AveiMil	16-Jul-10 1:18
comInterfaceProxyStub new	Neil Sl	23-Feb-10 6:37
Re: comInterfaceProxyStub new	Neil Sl	25-Feb-10 12:46
Re: comInterfaceProxyStub new	wladek5	23-Apr-11 8:08
ProgID? new	Member 3631625	3-Feb-10 3:07
Re: ProgID? new	Cristian Adam	5-Feb-10 1:46
Re: ProgID? new	Vladimir Rutsky	25-Jan-13 3:06
assemblyIdentity element name attribute seems to be wrong. new	Felix Collins	1-Oct-09 13:27
regsvr42 GUI new	stanX	31-Aug-09 2:52
Re: regsvr42 GUI new	Cristian Adam	31-Aug-09 5:47
Re: regsvr42 GUI new	stanliritche	3-Nov-11 5:01
Can not get the <comclass> info</comclass> new	fcfs	31-Jul-09 1:28
Can't get manifest to work new	Randem06	14-Jul-09 22:07
Re: Can't get manifest to work new	Cristian Adam	15-Jul-09 0:34
Re: Can't get manifest to work new	Randem06	15-Jul-09 1:21
Re: Can't get manifest to work new	Cristian Adam	15-Jul-09 3:40

Article

Browse Code

Stats

Revisions

Alternatives (1)

Comments (75)

Add your own alternative version

Tagged as

VC8.0

WinXP

C++

Windows

Dev

COM

Intermediate

VC9.0

Related Articles

regsvr42: Generate SxS Manifest Files from Native DLLs for Registration-Free COM

Generating IDL and Manifest Files with

Re: Can't get manifest to work new	Randem06	16-Jul-09 23:29
an't get manifest to work [modified] new	Randem06	16-Jul-09 23:59
Re: Can't get manifest to work new	Randem06	17-Jul-09 0:21
Re: Can't get manifest to work new	Cristian Adam	17-Jul-09 1:12
Re: Can't get manifest to work new	Randem06	17-Jul-09 1:50
Re: Can't get manifest to work new	Cristian Adam	17-Jul-09 3:02
Re: Can't get manifest to work new	Randem06	17-Jul-09 21:31
Re: Can't get manifest to work new	Vlad Vissoultchev	25-Sep-09 0:31
-version has no effect new	**Randem06**	**14-Jul-09 0:38**
Re: -version has no effect new	Cristian Adam	14-Jul-09 1:27
Re: -version has no effect new	Randem06	14-Jul-09 19:03
Thank you very much, Cristian! new	**Wolfgang Riedmann**	**30-May-09 4:06**
Question about 'comInterfaceExternalProxyStub' new	**pkooman3**	**26-May-09 6:51**

Last Visit: 31-Dec-99 20:00 Last Update: 20-Jun-15 21:46 Refresh 1 2 Next »

General News Suggestion Question Bug Answer Joke Rant Admin

Use Ctrl+Left/Right to switch messages, Ctrl+Up/Down to switch threads, Ctrl+Shift+Left/Right to switch pages.

Article

Browse Code

Stats

Revisions

Alternatives (1)

Comments (75)

Add your own alternative version

Tagged as

VC8.0

WinXP

C++

Windows

Dev

COM

Intermediate

VC9.0

Related Articles

regsvr42: Generate SxS Manifest Files from Native DLLs for Registration-Free COM

Generating IDL and Manifest Files with

Permalink | Advertise | Privacy | Terms of Use | Mobile

Layout: fixed | fluid

Web04 | 2.8.150618.2 | Last Updated 19 Aug 2008

Select Language ▼

Article Copyright 2008 by Cristian Adam
Everything else Copyright © CodeProject, 1999-2015

Article

Browse Code

Stats

Revisions

Alternatives (1)

Comments (75)

Add your own alternative version

Tagged as

VC8.0

WinXP

C++

Windows

Dev

COM

Intermediate

VC9.0

Related Articles

regsvr42: Generate SxS Manifest Files from Native DLLs for Registration-Free COM

Generating IDL and Manifest Files with