Cleanly Uninstalling Stubborn SQL Server Components

By: Aaron Bertrand | Read Comments (24) | Related Tips: More > Install and Uninstall

Problem

There are scenarios where SQL Server is difficult or impossible to uninstall, upgrade, or replace (and these can block you from installing a new version or using a specific named or default instance):

1. **An expired Evaluation Edition:**

```
Evaluation period has expired. For information on how to upgrade your evaluation software
please go to http://www.microsoft.com/sql/howtobuy
```

2. **An expired Management Studio:**

```
Your Microsoft SQL Server Management Studio evaluation period has expired.

You can get a key to extend your trial by registering this copy of Microsoft Management Studio
online. You may also purchase a key to activate the product.
```

3. **Unsupported operating system (after an OS upgrade):**

```
The operating system on this computer does not meet the minimum requirements for SQL Server
xxxx.
```

4. **Missing MSI files, e.g.:**

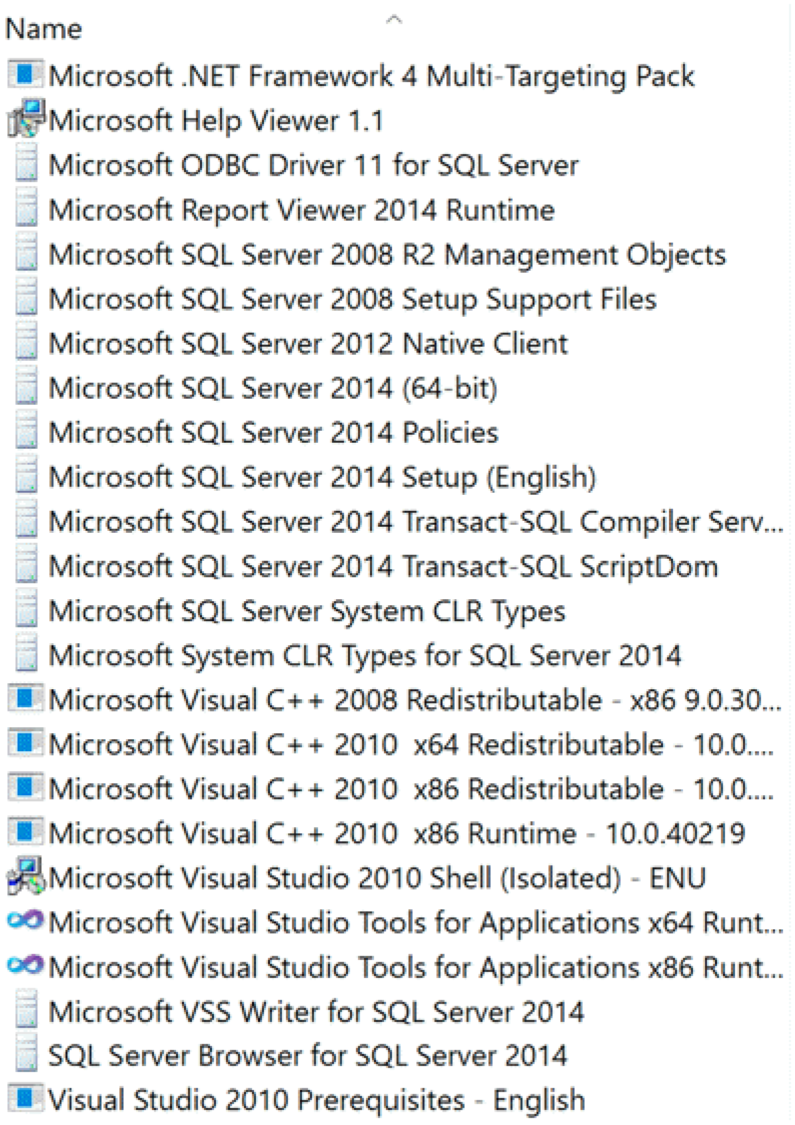
```
Slp: Target package: "C:\...\sql_engine_core_inst.msi"
Slp: InstallPackage: MsiInstallProduct returned the result code 2.
```

5. **Too many instances to remove individually, or other various errors during uninstall.**

In these situations, is there a way to cleanly uninstall SQL Server?

Solution

There is a built-in command called `msiexec` which has an uninstall parameter (`-x`). This command can be used to remove stubborn programs through brute force. First, though, you need to get an inventory of the GUIDs that represent the programs you need to remove. All you see in the Control Panel are the friendly names, as you can see here:



Name
Microsoft .NET Framework 4 Multi-Targeting Pack
Microsoft Help Viewer 1.1
Microsoft ODBC Driver 11 for SQL Server
Microsoft Report Viewer 2014 Runtime
Microsoft SQL Server 2008 R2 Management Objects
Microsoft SQL Server 2008 Setup Support Files
Microsoft SQL Server 2012 Native Client
Microsoft SQL Server 2014 (64-bit)
Microsoft SQL Server 2014 Policies
Microsoft SQL Server 2014 Setup (English)
Microsoft SQL Server 2014 Transact-SQL Compiler Serv...
Microsoft SQL Server 2014 Transact-SQL ScriptDom
Microsoft SQL Server System CLR Types
Microsoft System CLR Types for SQL Server 2014
Microsoft Visual C++ 2008 Redistributable - x86 9.0.30...
Microsoft Visual C++ 2010 x64 Redistributable - 10.0....
Microsoft Visual C++ 2010 x86 Redistributable - 10.0....
Microsoft Visual C++ 2010 x86 Runtime - 10.0.40219
Microsoft Visual Studio 2010 Shell (Isolated) - ENU
Microsoft Visual Studio Tools for Applications x64 Runt...
Microsoft Visual Studio Tools for Applications x86 Runt...
Microsoft VSS Writer for SQL Server 2014
SQL Server Browser for SQL Server 2014
Visual Studio 2010 Prerequisites - English

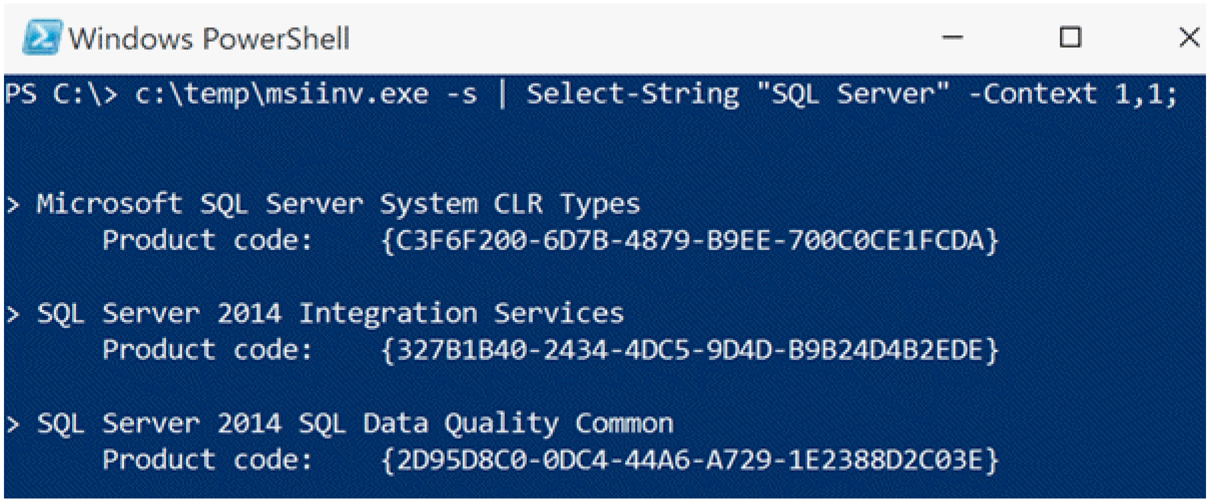
(This is a brand new system where I installed multiple SQL Server 2014 components in order to demonstrate my approach of cleanly removing them without using the Control Panel or SQL Server's own setup, but rather using `msiexec` and the GUIDs for the products.)

You can get the GUID associated with each installed product by searching around the registry, but I would rather use the MSI Inventory tool, which you can download (`msiinv_new.zip`) from this OneDrive folder, and extract `msiinv.exe` to a folder, say `C:\temp\`.

Once that file is there, you can open a PowerShell console, and run the following code:

```
c:\temp\msiinv.exe -s | Select-String "SQL Server" -Context 0,1
```

Here is a truncated version of what my output from that command looks like (there is a benign error you can ignore):



```
PS C:\> c:\temp\msiinv.exe -s | Select-String "SQL Server" -Context 1,1;


> Microsoft SQL Server System CLR Types
      Product code:    {C3F6F200-6D7B-4879-B9EE-700C0CE1FCDA}

> SQL Server 2014 Integration Services
      Product code:    {327B1B40-2434-4DC5-9D4D-B9B24D4B2EDE}

> SQL Server 2014 SQL Data Quality Common
      Product code:    {2D95D8C0-0DC4-44A6-A729-1E2388D2C03E}
```

This output is marginally useful for now; I can quickly scan the list and see all the GUID codes for the products I want to remove, and start mentally filtering any out as necessary. I could manually copy and change the output to feed each GUID individually to `msiexec`, but ultimately where I want to end up is a set of commands I can run directly from the command line, for example:

```
rem Microsoft SQL Server System CLR Types
msiexec /x "{C3F6F200-6D7B-4879-B9EE-700C0CE1FCDA}"

rem SQL Server 2014 Integration Services
msiexec /x "{327B1B40-2434-4DC5-9D4D-B9B24D4B2EDE}"

rem SQL Server 2014 SQL Data Quality Common
msiexec /x "{2D95D8C0-0DC4-44A6-A729-1E2388D2C03E}"

...
```

And I get there with a little help from my good buddy, co-worker, and PowerShell officionado, Allen White (@SQLRunr):

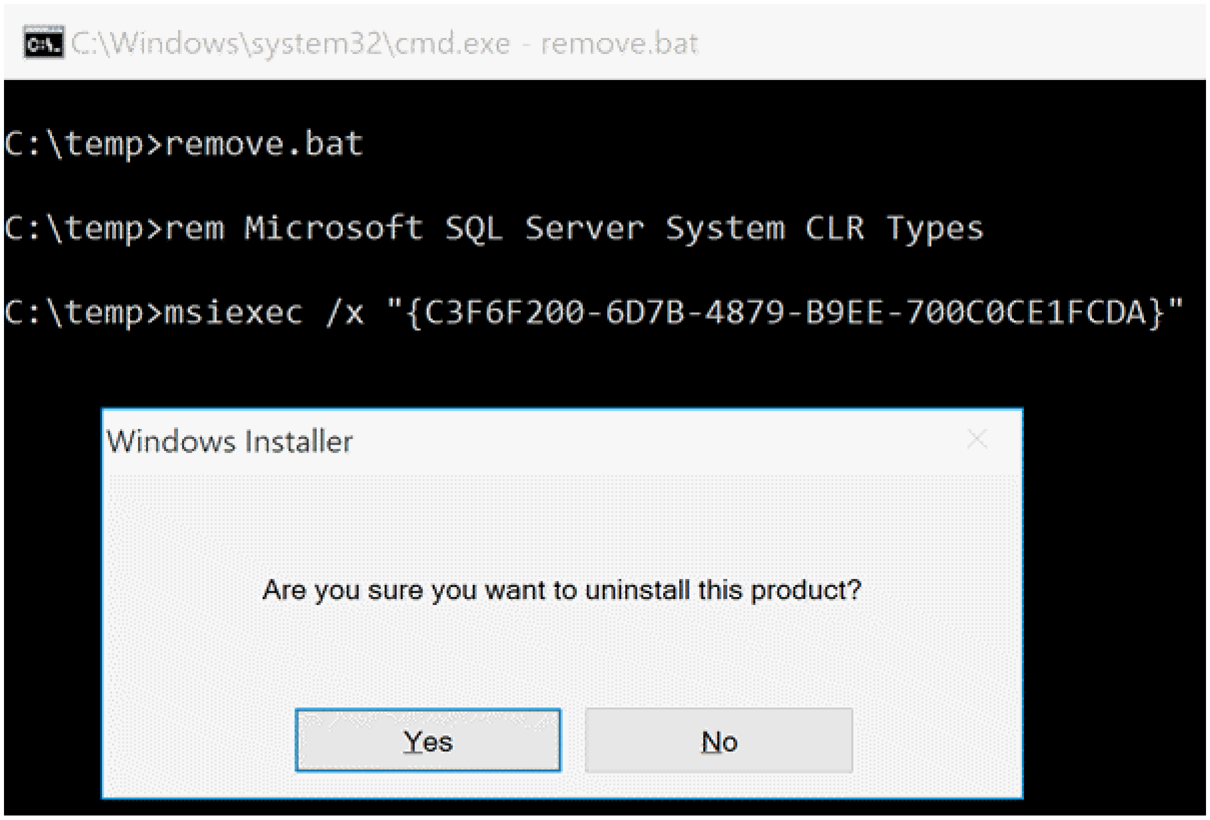
```
$a = c:\temp\msiinv.exe -s | Select-String "SQL Server" -Context 1,1;
$a = $a -replace "Product code: ","msiexec /x """;
$a = $a -replace ">", "rem";
$a = $a -replace "\t", "";
$a = $a -replace "}","}""";
$a | Out-File c:\temp\remove.bat -encoding ascii;
```

Now, you can open up `remove.bat`, look through the list, and remove any products you *don't* want to uninstall. The format will look like this, almost exactly how I wanted them:

```
rem Microsoft SQL Server System CLR Types
  msiexec /x "{C3F6F200-6D7B-4879-B9EE-700C0CE1FCDA}"

rem SQL Server 2014 Integration Services
  msiexec /x "{327B1B40-2434-4DC5-9D4D-B9B24D4B2EDE}"

rem SQL Server 2014 SQL Data Quality Common
```

```
  msiexec /x "{2D95D8C0-0DC4-44A6-A729-1E2388D2C03E}"

...
```

Now run remove.bat from the command line. You'll get prompts like this:



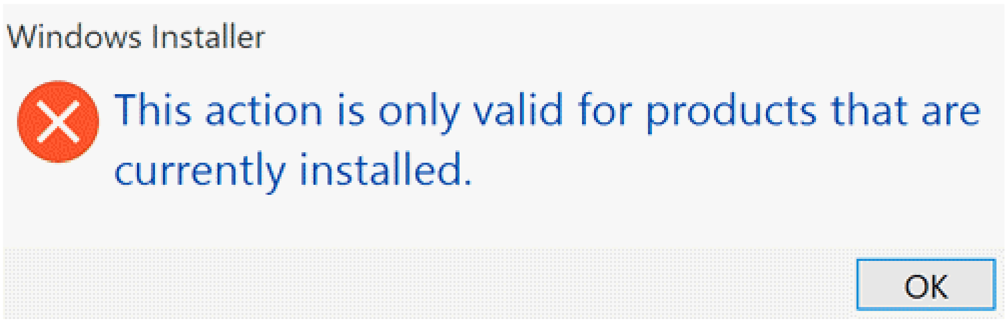
```
C:\temp>remove.bat

C:\temp>rem Microsoft SQL Server System CLR Types

C:\temp>msiexec /x "{C3F6F200-6D7B-4879-B9EE-700C0CE1FCDA}"
```

Are you sure you want to uninstall this product?

And you'll also likely see some confirmation dialogs from User Account Control. You may get some errors or dependency warnings, and to avoid some of the prompts, you'll want to make sure all related services have been manually stopped. You might want to run from an elevated command prompt (perhaps even using the "super admin" if you're on Windows 10), and also play with some of the switches to `msiexec`, like `/quiet`. You can enable `/quiet` by changing the second last line in the Powershell script above to:

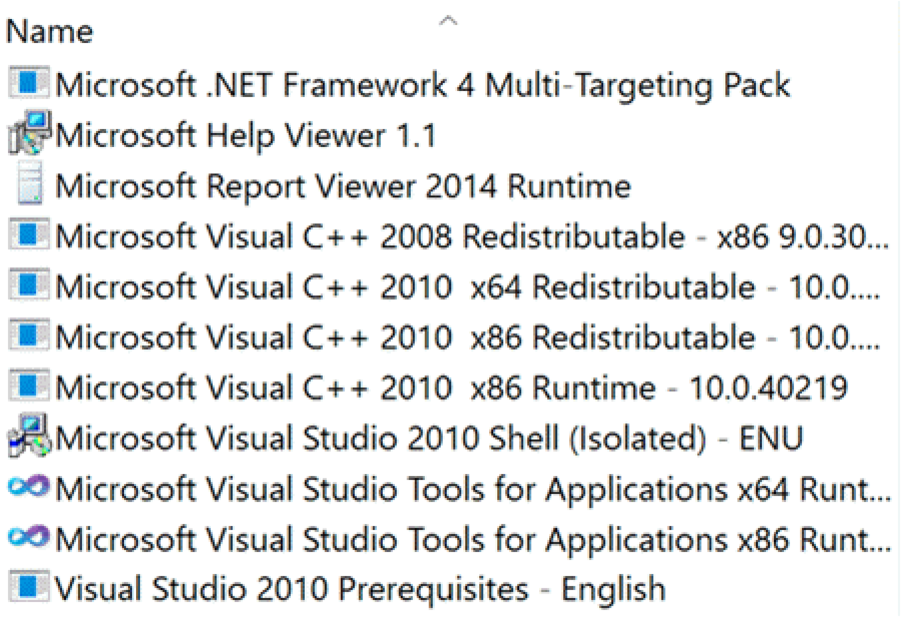
```
$a = $a -replace "}","}"" /quiet";
```

Also, you may have to run the script twice to completely remove things that failed the first time due to dependency order. The second time around, if you don't use `/quiet`, you will see multiple notifications that the product you're trying to remove is no longer installed:



This action is only valid for products that are currently installed.

But after running through it twice, this should leave you with a much cleaner Control Panel:


Name
Microsoft .NET Framework 4 Multi-Targeting Pack
Microsoft Help Viewer 1.1
Microsoft Report Viewer 2014 Runtime
Microsoft Visual C++ 2008 Redistributable - x86 9.0.30...
Microsoft Visual C++ 2010 x64 Redistributable - 10.0....
Microsoft Visual C++ 2010 x86 Redistributable - 10.0....
Microsoft Visual C++ 2010 x86 Runtime - 10.0.40219
Microsoft Visual Studio 2010 Shell (Isolated) - ENU
Microsoft Visual Studio Tools for Applications x64 Runt...
Microsoft Visual Studio Tools for Applications x86 Runt...
Visual Studio 2010 Prerequisites - English

The remainder of these items should really be much easier to remove manually; or, you may not want to bother; or, you could repeat the process above with different arguments to `Select-String`. I wanted to focus on a single run-through with the SQL Server components, since they are the ones that prove most problematic.

Summary

This is not the most intuitive approach in the world, but I hope it helps some of you remove stubborn, sticky SQL Server components from your systems. It constitutes a bit more work than normal uninstall operations, but when Control Panel or SQL Server Setup won't cooperate, this might be the next best thing. (Note that this procedure will not be effective in completely removing a clustered instance of SQL Server; I wouldn't even try.)

Next Steps

- Try this approach if you have a system with SQL Server components that have expired or can't be removed in the usual way.
- See these related tips and other resources:
  - SQL Server 2008 Service Pack 1 (SP1) Uninstall
  - Uninstalling a SQL Server Clustered Instance
  - Removing the SQL Server Management Data Warehouse
  - Fun with software : uninstalling SQL Server 2008 R2 Evaluation Edition
  - All Install and Uninstall Tips

Last Update: 2015-10-06

About the author

Aaron Bertrand is a Senior Consultant at SQL Sentry, Inc., and has been contributing to the community for **about two decades**, first earning the Microsoft MVP award in 1997.

View all my tips

Related Resources

- More SQL Server DBA Tips...