**https://learn.microsoft.com/en-us/troubleshoot/sql/install/restore-missing-windows-installer-cache-files**

**Solution 3: Use the FindSQLInstalls.vbs script**

To complete the steps in this procedure, you have to copy the FindSQLInstalls.vbs script in the *FixMissingMSI* folder from the [GitHub repository](https://github.com/suyouquan/SQLSetupTools/tree/master/FixMissingMSI) to a local folder on the computer where you are trying to update your SQL Server installation.

**Note**

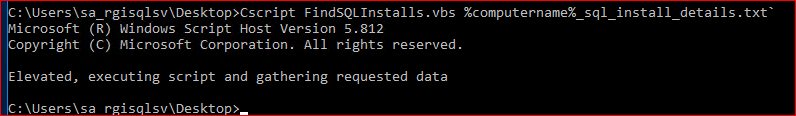
The FindSQLInstalls.vbs script collects package information to correct invalid package paths. This script is used against the source locations to make sure that all MSP packages are in the Windows Installer cache directory. After executing the commands indicated in the Action needed lines in the script output file, the missing packages will be re-added if the original source media is available.

To resolve these problems by using a script, follow these steps:

1. Go to the [FindSQLInstalls.vbs raw page on GitHub](https://raw.githubusercontent.com/suyouquan/SQLSetupTools/master/FixMissingMSI/FindSQLInstalls.vbs).
2. Copy all the contents on the page to a new text file.
3. Save the text file as *FindSQLInstalls.vbs*.
4. Open an elevated Command Prompt to the directory to which you saved the *FindSQLInstalls.vbs* file, and run the command:

ConsoleCopy

Cscript FindSQLInstalls.vbs %computername%\_sql\_install\_details.txt`.



1. Open the file from step 2 in a text editor such as Notepad, and identify the problems that cause the failure. To do this, search the text file for string patterns such as the following:
   * do not
   * !!!
2. Based on the results in the previous step, take the steps that are required.

**Note**

Look for more information about these steps in the [**Examples**](https://learn.microsoft.com/en-us/troubleshoot/sql/install/restore-missing-windows-installer-cache-files#findsqlinstallsvbs-examples) section.

1. Repeat steps 2 through 4 until the text file that is created in step 2 no longer contains text that references invalid paths or missing files for the component that is being updated.

<https://raw.githubusercontent.com/suyouquan/SQLSetupTools/master/FixMissingMSI/FindSQLInstalls.vbs>

' Copyright © Microsoft Corporation. All Rights Reserved.

' This code released under the terms of the

' Microsoft Public License (MS-PL, http://opensource.org/licenses/ms-pl.html.)

On Error Resume Next

Dim arrSubKeys, arrSubKeys2

Dim objFSO, objShell, objFile, objReg, objConn, objExec

Dim strComputer, strKeyPath, strNewSource

Dim strWorkstationName, strDBPath, strSubKey, strSubKey2(), strKeyPath02, strRetValue00

Dim strRetValue01, strRetValue02, strRetValNew02, strRetValNew03, strRetValNew04, strRetValNew05, strRetValNew06, strRetValNew07, strRetValNew08, strRetValNew09, strRetValue10, strRetValNew10, strRetValNew11, strRetValNew12, strRetValNew13, strRetValNew14, strRetValNew14a, strRetValNew14b, strRetValNew15, strRetValNew15a, strRetValNew15b, strRetValNew16, strRetValNew17, strRetValNew18

Const HKCR = &H80000000 'HKEY\_CLASSES\_ROOT

Const HKLM = &H80000002 'HKEY\_LOCAL\_MACHINE

Const ForReading = 1, ForWriting = 2, ForAppEnding = 8

' Checking for Elevated permissions

Dim oShell, oExec

szStdOutszStdOut = ""

Set oShell = CreateObject("WScript.Shell")

Set oExec = oShell.Exec("whoami /groups")

Do While (oExec.Status = cnWshRunning)

WScript.Sleep 100

if not oExec.StdOut.AtEndOfStream Then

szStdOut = szStdOut & oExec.StdOut.ReadAll

end If

Loop

select case oExec.ExitCode

case 0

if not oExec.StdOut.AtEndOfStream Then

szStdOut = szStdOut & oExec.StdOut.ReadAll

End If

If instr(szStdOut,"Mandatory Label\High Mandatory Level") Then

wscript.echo "Elevated, executing script and gathering requested data"

Else

if instr(szStdOut,"Mandatory Label\Medium Mandatory Level") Then

Wscript.echo "Not Elevated must run from Administrative commmand line."

Else

Wscript.echo "Gathering requested data..."

end If

End If

case Else

if not oExec.StdErr.AtEndOfStream Then

wscript.echo oExec.StdErr.ReadAll

end If

end select

'

' Leaving strNewSource will result in no search path updating.

' Currently DO NOT EDIT these.

strNewSource = ""

strNewRTMSource = ""

' Define string values

strComputer = "."

strSQLName = "SQL"

strDotNetName = ".NET"

strVStudioName = "Visual Studio"

strXML = "XML"

strOWC = "Microsoft Office 2003 Web Components"

strKeyPath = "Installer\Products"

strKeyPath2 = "SOFTWARE\Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\Products"

strNValue00 = "ProductName"

strNValue01 = "PackageName"

strNValue02 = "LastUsedSource"

strNValue03 = "InstallSource"

strNValue04 = "LocalPackage"

strNValue05 = "DisplayVersion"

strNValue06 = "InstallDate"

strNValue07 = "UninstallString"

strNValue08 = "PackageCode"

strNValue09 = "MediaPackage"

strNValue10 = "InstallSource"

strNValue11 = "AllPatches"

strNValue12 = "NoRepair"

strNValue13 = "MoreInfoURL"

strNValue14 = "PackageName"

strNValue15 = "LastUsedSource"

strNValue16 = "Uninstallable"

strNValue17 = "DisplayName"

strNValue18 = "Installed"

If WScript.arguments.count <> 1 Then

WScript.echo "Usage: cscript " & WScript.scriptname & " outputfilename.txt"

WScript.quit

End If

'--Setup the output file

Set fso = CreateObject("Scripting.FileSystemObject")

Set txtFile = fso.OpenTextFile(WScript.arguments(0), ForWriting, True)

If err.number <> 0 Then

WScript.echo "Error 0x" & myHex(err.number,8) & ": " & err.source & " - " & err.description

WScript.quit

End If

txtFile.writeline "Products installed on the local system"

txtFile.writeline " "

txtFile.writeline " "

Set objFSO = CreateObject("Scripting.FileSystemObject")

Set objShell = WScript.CreateObject("WScript.Shell")

'--Set up the registry provider.

Set objReg = GetObject("winmgmts:\\" & strComputer & \_

"\root\default:StdRegProv")

Set wiInstaller = CreateObject("WindowsInstaller.Installer")

'--Enumerate the "installer\products" key on HKCR

objReg.EnumKey HKCR, strKeyPath, arrSubKeys

For Each strSubKey In arrSubKeys

' Define the various registry paths

strProduct01 = "Installer\Products\" & strSubKey

strKeyPath02 = "Installer\Products\" & strSubKey & "\SourceList"

strKeyPath03 = "Installer\Products\" & strSubKey & "\SourceList\Media"

strInstallSource = "SOFTWARE\Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\Products\" & strSubKey & "\InstallProperties\"

strInstallSource2 = "SOFTWARE\Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\Products\" & strSubKey & "\patches\"

strInstallSource3 = "SOFTWARE\Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\Patches"

strInstallSource5 = "SOFTWARE\Classes\Installer\Patches\"

strInstallSource6 = "SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\"

strInstallSource7 = "SOFTWARE\Microsoft\Microsoft SQL Server\"

strInstallSource8 = "SOFTWARE\Wow6432Node\Microsoft\Microsoft SQL Server\"

' Pull the intial values

objReg.GetStringValue HKCR, strProduct01, strNValue00, strRetValue00

objReg.GetStringValue HKCR, strKeyPath02, strNValue01, strRetValue01

objReg.GetStringValue HKCR, strKeyPath02, strNValue02, strRetValue02

strRetValNew02 = Mid(strRetValue02, 5)

objReg.GetStringValue HKCR, strKeyPath03, strNValue09, strRetValue09

strRetValue10 = strNewRTMSource & strRetValue09

objReg.GetStringValue HKLM, strInstallSource, strNValue03, strRetValNew03

objReg.GetStringValue HKLM, strInstallSource, strNValue04, strRetValNew04

objReg.GetStringValue HKLM, strInstallSource, strNValue05, strRetValNew05

objReg.GetStringValue HKLM, strInstallSource, strNValue06, strRetValNew06

objReg.GetStringValue HKLM, strInstallSource, strNValue07, strRetValNew07

objReg.GetStringValue HKLM, strInstallSource, strNValue10, strRetValNew10

objReg.GetStringValue HKLM, strInstallSource, strNValue12, strRetValNew12

objReg.GetStringValue HKLM, strInstallSource, strNValue13, strRetValNew13

objReg.GetStringValue HKLM, strInstallSource2, strNValue11, strRetValNew11

' Pull the Product Code from the Uninstall String

strProdCode = strRetValNew07

ProdCodeLen = Len(strProdCode)

ProdCodeLen = ProdCodeLen - 14

strRetValNew08 = Right(strProdCode, ProdCodeLen)

' Pull out path from LastUsedSource

strGetRealPath = strRetValue02

GetRealPath = Len(strRetValue02)

strRealPath = Mid(strRetValue02, 5, GetRealPath)

' Identifie the string in the ProductName

If instr(1, strRetValue00, strSQLName, 1) Then

' Start the log output

txtFile.writeline "================================================================================"

txtFile.writeline "PRODUCT NAME : " & strRetValue00

txtFile.writeline "================================================================================"

txtFile.writeline " Product Code: " & strRetValNew08

txtFile.writeline " Version : " & strRetValNew05

txtFile.writeline " Most Current Install Date: " & strRetValNew06

txtFile.writeline " Target Install Location: " & strRetValNew13

txtFile.writeline " Registry Path: "

txtFile.writeline " HKEY\_CLASSES\_ROOT\" & strKeyPath02

txtFile.writeline " Package : " & strRetValue01

txtFile.writeline " Install Source: " & strRetValue10

txtFile.writeline " LastUsedSource: " & strRetValue02

' txtFile.writeline "Does this file on this path exist? " & strRetValNew02 & "\" & strRetValue01

If fso.fileexists(strRetValNew02 & "\" & strRetValue01) Then

txtFile.writeline " "

txtFile.writeline " " & strRetValue01 & " exists on the LastUsedSource path, no actions needed."

Else

txtFile.writeline " "

txtFile.writeline " !!!! " & strRetValue01 & " DOES NOT exist on the path in the path " & strRealPath & " !!!!"

txtFile.writeline " "

txtFile.writeline " Action needed, re-establish the path to " & strRealPath

' Placeholder for altering the LastUsedSource by adding source location and Forcing search of list

' If strNewSource <> "" Then

' txtFile.writeline " New Install Source Path Added: " & strNewSource

' wiInstaller.AddSource strRetValNew08, "", strNewSource

' Else

' If strNewRTMSource <> "" Then

' wiInstaller.AddSource strRetValNew08, "", strNewRTMSource

' txtFile.writeline " Forcing SourceList Resolution For: " & strRetValNew08

' wiInstaller.ForceSourceListResolution strRetValNew08, ""

' End If

' End If

End If

txtFile.writeline " "

txtFile.writeline "Installer Cache File: " & strRetValNew04

If fso.fileexists(strRetValNew04) Then

txtFile.writeline " "

txtFile.writeline " Package exists in the Installer cache, no actions needed."

txtFile.writeline " Any missing packages will update automatically if needed assuming that"

txtFile.writeline " the LastUsedSource exists."

txtFile.writeline " "

txtFile.writeline " Should you get errors about " & strRetValNew04 & " or " & strRealPath & strRetValue01 & " then you"

txtFile.writeline " may need to manually copy the file, if file exists replace the problem file, "

txtFile.writeline " Copy and paste the following command line into an administrative command prompt:"

txtFile.writeline " "

txtFile.writeline " Copy " & chr(34) & strRealPath & strRetValue01 & chr(34) & " " &strRetValNew04

txtFile.writeline " "

ElseIf fso.fileexists(strRetValNew02 & "\" & strRetValue01) Then

fso.CopyFile strRetValNew02 & "\" & strRetValue01, strRetValNew04

If fso.fileexists(strRetValNew04) Then

txtFile.writeline " "

txtFile.writeline " Missing cache file replaced by copying " & strRealPath & strRetValue01 & " to " & strRetValNew04

txtFile.writeline " Previously missing package " & strRetValNew04 & " now exists in the Installer cache."

txtFile.writeline " "

End If

Else

txtFile.writeline " "

txtFile.writeline " !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

txtFile.writeline " !!!! " & strRetValNew04 & " DOES NOT exist in the Installer cache. !!!!"

txtFile.writeline " !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

txtFile.writeline " "

txtFile.writeline " Action needed, recreate or re-establish path to the directory:"

txtFile.writeline " " & strRealPath & "then rerun this script to update installer cache and results"

txtFile.writeline " The path on the line above must exist at the root location to resolve"

txtFile.writeline " this problem with your msi/msp file not being found or corrupted,"

txtFile.writeline " In some cases you may need to manually copy the missing file or manually"

txtFile.writeline " replace the problem file overwriting it is exist: "

txtFile.writeline " "

txtFile.writeline " Copy " & chr(34) & strRealPath & strRetValue01 & chr(34) & " " &strRetValNew04

txtFile.writeline " "

txtFile.writeline " Replace the existing file if prompted to do so."

txtFile.writeline " "

End If

txtFile.writeline " "

txtFile.writeline strRetValue00 & " Patches Installed "

txtFile.writeline "--------------------------------------------------------------------------------"

err.clear

objReg.EnumKey HKLM, strInstallSource2, arrSubKeys2

uUpperBounds = UBound(arrSubKeys2,1)

If err.number = 0 Then

For Each strSubKey2 in arrSubKeys2

' WScript.echo "value = " & strSubKey2

strKeyPath04 = "Installer\Patches\" & strSubKey2 & "\SourceList"

objReg.GetDWORDValue HKLM, strInstallSource2 & "\" & strSubKey2 & "\", strNValue16, strRetValue16

objReg.GetStringValue HKCR, strKeyPath04, strNValue15, strRetValue15a

objReg.GetStringValue HKCR, strKeyPath04, strNValue14, strRetValue14a

objReg.GetStringValue HKCR, strKeyPath02, strNValue15, strRetValue15b

objReg.GetStringValue HKCR, strKeyPath02, strNValue14, strRetValue14b

objReg.GetStringValue HKLM, strInstallSource2 & "\" & strSubKey2 & "\", strNValue17, strRetValue17

objReg.GetStringValue HKLM, strInstallSource2 & "\" & strSubKey2 & "\", strNValue18, strRetValue18

objReg.GetStringValue HKLM, strInstallSource2 & "\" & strSubKey2 & "\", strNValue13, strRetValue13a

objReg.GetStringValue HKLM, strInstallSource3 & "\" & strSubKey2 & "\", strNValue04, strRetValue04a

' Pull the URL from the MoreInfoURL String

strMoreInfoURL = strRetValue13a

MoreInfoURLLen = Len(strMoreInfoURL)

strRetValue13b = Right(strMoreInfoURL, 42)

' Pull the URL from the LastUsedPath String

strLastUsedPath = strRetValue15a

LastUsedPathLen = Len(strLastUsedPath)

'LastUsedPathLen = LastUsedPathLen - 15

strRetValue15c = Mid(strLastUsedPath, 5)

txtFile.writeline " Display Name: " & strRetValue17

txtFile.writeline " KB Article URL: " & strRetValue13b

txtFile.writeline " Install Date: " & strRetValue18

txtFile.writeline " Uninstallable: " & strRetValue16

txtfile.writeline " Patch Details: "

txtFile.writeline " HKEY\_CLASSES\_ROOT\Installer\Patches\" & strSubKey2

txtFile.writeline " PackageName: " & strRetValue14a

' Determine if someone has modified the Uninstallable state from 0 to 1 allowing possible unexpected uninstalls

txtFile.writeline " Patch LastUsedSource: " & strRetValue15a

txtFile.writeline " Installer Cache File Path: " & strRetValue04a

txtFile.writeline " Per " & strInstallSource3 & "\" & strSubKey2 & "\" & strNValue04

mspFileName = (strRetValue15c & strRetValue14a)

If strRetValue14a <> "" Then

If fso.fileexists(strRetValue04a) Then

txtFile.writeline " "

txtFile.writeline " Package exists in the Installer cache, no actions needed."

txtFile.writeline " Package will update automatically if needed assuming that"

txtFile.writeline " the LastUsedSource exists."

txtFile.writeline " "

txtFile.writeline " Should you get errors about " & strRetValue04a & " or " & strRetValue15c & strRetValue14a & " then you"

txtFile.writeline " may need to manually copy missing files, if file exists replace the problem file, "

txtFile.writeline " Copy and paste the following command line into an administrative command prompt."

txtFile.writeline " "

txtFile.writeline " Copy " & chr(34) & strRetValue15c & strRetValue14a & chr(34) & " " & strRetValue04a

txtFile.writeline " "

ElseIf fso.fileexists(mspFileName) Then

fso.CopyFile mspFileName, strRetValue04a

If fso.fileexists(strRetValue04a) Then

txtFile.writeline " "

txtFile.writeline " Missing cache file replaced by copying " & strRetValue15c & strRetValue14a & " to " & strRetValue04a

txtFile.writeline " Previously missing package " & strRetValNew04 & " now exists in the Installer cache."

txtFile.writeline " "

End If

' End If

Else

txtFile.writeline " "

txtFile.writeline "!!!! " & strRetValue04a & " package DOES NOT exist in the Installer cache. !!!!"

txtFile.writeline " "

txtFile.writeline " Action needed, recreate or re-establish path to the directory:"

txtFile.writeline " " & strRetValue15c & " then rerun this script to update installer cache and results"

txtFile.writeline " The path on the line above must exist at the root location to resolve"

txtFile.writeline " this problem with your msi/msp file not being found or corrupted,"

txtFile.writeline " In some cases you may need to manually copy missing files or manually"

txtFile.writeline " replace the problem file, "

txtFile.writeline " "

txtFile.writeline " Copy " & chr(34) & strRetValue15c & strRetValue14a & chr(34) & " " & strRetValue04a

txtFile.writeline " "

txtFile.writeline " Replace the existing file if prompted to do so."

txtFile.writeline " "

txtFile.writeline " Use the following URL to assist with downloading the patch:"

txtFile.writeline " " & strRetValue13b

txtFile.writeline " "

txtFile.writeline " "

End If

Else

txtFile.writeline " "

End If

next

Else

txtfile.writeline " "

txtfile.Writeline " No Patches Found"

txtfile.writeline " "

End If

End If

Next

txtFile.Close

Set txtFile = Nothing

Set fso = Nothing

Copy "H:\x64\Setup\sql\_engine\_core\_inst\_msi\sql\_engine\_core\_inst.msi" F:\Missing\_files\f5e6292.msi

Copy "Source file path with filename" Destination path with filename