

Installing all Microsoft Visual C++ Runtime modules

On my blog post [Latest Supported Visual C++ Downloads](#) you a link to the Microsoft site [Latest Supported Visual C++ Downloads](#). Once downloaded, how to install all these updates?

I made a script that installs all the Microsoft C++ Runtime modules. On x86 only the x86 modules are installed. On x64, both x86 and x64 modules are installed.

The list has been populated on June, 14th, 2014. So the list of C++ Runtime modules might – and will be – change.

The folder list should be as follows:

```
Folder PATH listing for volume OS
Volume serial number is 000000C0 0475:818B
C:.
|   install_all_VisualCPlusPlusRuntimes_v101.vbs
|
+---01.Microsoft Visual C++ 2005 SP1 Redistributable Package
|       vcredist_x64.exe
|       vcredist_x86.exe
|
+---02.Microsoft Visual C++ 2005 SP1 Redistributable Package
ATL Security Update
|       vcredist_x64.exe
|       vcredist_x86.exe
|
+---03.Microsoft Visual C++ 2005 SP1 Redistributable Package
MFC Security Update
|       vcredist_x64.EXE
|       vcredist_x86.EXE
|
+---04.Microsoft Visual C++ 2008 SP1 Redistributable Package
|       vcredist_x64.exe
```

```
|         vcredist_x86.exe
|
+---05.Microsoft Visual C++ 2008 SP1 Redistributable Package
MFC Security Update
|         vcredist_x64.exe
|         vcredist_x86.exe
|
+---06.Microsoft Visual C++ 2010 SP1 Redistributable Package
|         vcredist_x64.exe
|         vcredist_x86.exe
|
+---07.Microsoft Visual C++ 2010 SP1 Redistributable Package
MFC Security Update
|         vcredist_x64.exe
|         vcredist_x86.exe
|
+---08.Microsoft Visual C++ 2012 Update 3
|         vcredist_x64.exe
|         vcredist_x86.exe
|
+---09.Microsoft Visual C++ 2013 Redistributable Package
|         vcredist_x64.exe
|         vcredist_x86.exe
|
+---10.Microsoft Visual C++ 2015 Redistributable Update 3
|         vcredist_x64.exe
|         vcredist_x86.exe
\---11.Microsoft Visual C++ Redistributable for Visual Studio
2017
         vcredist_x64.exe
         vcredist_x86.exe
```

The only thing you have to do is to download all the redistributable packages:

- [Microsoft Visual C++ 2005 SP1 Redistributable Package \(x86\)](#)
- [Microsoft Visual C++ 2005 SP1 Redistributable Package \(x64\)](#)
- [Microsoft Visual C++ 2005 SP1 Redistributable Package](#)

- [ATL Security Update \(x86 and x64\)](#)
- [Microsoft Visual C++ 2005 SP1 Redistributable Package MFC Security Update \(x86 and x64\)](#)
- [Microsoft Visual C++ 2008 SP1 Redistributable Package \(x86\)](#)
- [Microsoft Visual C++ 2008 SP1 Redistributable Package \(x64\)](#)
- [Microsoft Visual C++ 2008 SP1 Redistributable Package MFC Security Update \(x86 and x64\)](#)
- [Microsoft Visual C++ 2010 SP1 Redistributable Package \(x86\)](#)
- [Microsoft Visual C++ 2010 SP1 Redistributable Package \(x64\)](#)
- [Microsoft Visual C++ 2010 SP1 Redistributable Package MFC Security Update \(x86 and x64\)](#)
- [Microsoft Visual C++ 2012 Update 3 \(x86 and x64\)](#)
- [Visual C++ Redistributable Packages for Visual Studio 2013](#)
- [Microsoft Visual C++ 2015 Redistributable Update 3 \(x86 and x64\)](#)
- [Microsoft Visual C++ Redistributable for Visual Studio 2017 \(x86 and x64\)](#)

The code:

```

,
=====
=====
' Install all Microsoft C++ Runtimes
' Created by Willem-Jan Vroom
' Version history:
,
' 0.0.1
'   Initial version
,
' 1.0.0
'   Final version
,

```

```
' 1.0.1
'     Added support for C++ Redistributable 2015
'
' 2.0.0
'     Addes support for Microsoft Visual C++ Redistributable
for Visual Studio 2017
'
```

```
=====
=====
```

```
' -----
-----
' Declare the most variables.
' -----
-----
```

Option Explicit

```
Dim objShell           : set objShell           =
WScript.CreateObject("WScript.Shell")
Dim objLogFileFS0      : Set objLogFileFS0      =
CreateObject("Scripting.FileSystemObject")
Dim objFS0             : Set objFS0             =
CreateObject("Scripting.FileSystemObject")
Dim objProcessEnv      : Set objProcessEnv      =
objShell.Environment("PROCESS")
Dim objWMIService      : Set objWMIService      =
GetObject("winmgmts:\\.\root\cimv2")
Dim fn_objWMIService   : Set fn_objWMIService   =
GetObject("winmgmts:\\.\root\cimv2")
Dim objReg             : Set objReg             =
GetObject("winmgmts:\\.\root\default:StdRegProv")
Dim objLogFile
Dim CurrentDir         : CurrentDir             =
Left(Wscript.ScriptFullname, InstrRev(Wscript.ScriptFullname,
"\"))
CurrentDir             =
Left(CurrentDir, len(CurrentDir) - 1)
Dim strcomputerName    : strcomputerName       =
objProcessEnv("COMPUTERNAME")
Dim strLogLocation     : strLogLocation        =
```

```
"C:\WINDOWS\system32\Logfiles"
```

```
Dim strOutputFile : strOutputFile =  
strLogLocation & "\" & strcomputerName &  
"_Installation_C++Runtimes_" & Replace(FormatDateTime(Now(),  
2),"/","-") & ".log"
```

```
Dim strArchitecture : strArchitecture = ""
```

```
Dim strCurrentUser : strCurrentUser = ""
```

```
Dim strCurrentUserSID : strCurrentUserSID = ""
```

```
Dim strOS : strOS = ""
```

```
Dim strCommand : strCommand = ""
```

```
Dim strLine : strLine = ""
```

```
Dim strQuery : strQuery = ""
```

```
Dim strCommonDesktop : strCommonDesktop = ""
```

```
Dim strArray : strArray = ""
```

```
Dim strAppDataFolder : strAppDataFolder = ""
```

```
Dim strProcess : strProcess = ""
```

```
Dim strValue : strValue = ""
```

```
Dim strKeyPath : strKeyPath = ""
```

```
Dim strValueName : strValueName = ""
```

```
Dim strLogFile
```

```
Dim dwValue : dwValue = 0
```

```
Dim arrFolderUnsorted : Redim arrFolderUnsorted(0)
```

```
Dim arrArguments
```

```
Dim arrLanguage
```

```
Dim valCounter
```

```
Dim valResult
```

```
Dim valReturnCode
```

```
Dim colProcess, objProcess
```

```
Dim valOSBuildNumber
```

```
Dim objSubfolder
```

```
Dim colItems, objItem, Subfolder
```

```
Dim colSoftware, objSoftware
```

```
Dim arrValues, strHolder
```

```
dim objFolders, objFolder, arrFolders, arrVersions,  
objVersion
```

```
Const ForWriting = 2
```

```
Const ForReading          = 1
Const ForAppending        = 8
Const OverWriteFiles      = True
const HKEY_CURRENT_USER   = &H80000001
const HKEY_LOCAL_MACHINE  = &H80000002
Const HKEY_USERS           = &H80000003
```

```
' -----
' -----
' Create the log location (if not exists)
' Open the logfile.
' -----
' -----
```

```
CreateFolderStructure(strLogLocation)
OpenLogFile()
WriteToLog "- ACTION: script started."
WriteToLog " "
```

```
' -----
' -----
' Detect the current OS.
' -----
' -----
```

```
Set colItems = objWMIService.ExecQuery("Select
Caption,BuildNumber from Win32_OperatingSystem")
For Each objItem in colItems
    strOS          = objItem.Caption
    valOSBuildNumber = objItem.BuildNumber
next
```

```
' -----
' -----
' Detect the current processor architecture.
' -----
' -----
```

```
if
objFS0.FolderExists(objShell.ExpandEnvironmentStrings("%windir
%") & "\SysWOW64\Config") then
```

```

    strArchitecture = "x64"
    arrVersions=Array("x86","x64")
    else
    strArchitecture = "x86"
    arrVersions=Array("x86")
end if

```

```

    WriteToLog("##### Details regarding operating system
#####")

```

```

    WriteToLog("Found Operating System:      " & strOS)

```

```

    WriteToLog("Found architecture:          " &
strArchitecture)

```

```

    WriteToLog("##### End details regarding operating
system          #####")

```

```

    WriteToLog(" ")

```

```

        WriteToLog("##### Installation
#####")

```

```

' -----
-----
' Sets SEE_MASK_NOZONECHECKS to 1.
' That will avoid a screen that asks for permission to run an
application from
' a share or mapped drive.
' -----
-----

```

```

objProcessEnv("SEE_MASK_NOZONECHECKS") = 1

```

```

' -----
-----
' Start the installation:
' 1. Loop throu all the subfolders, sorted.
'     Source:

```

```

http://blogs.technet.com/b/heyscriptingguy/archive/2004/11/22/how-can-i-list-the-members-of-a-group-in-alphabetical-order.aspx

```

```

' 2. Depending on the foldername the correct parameters are
passed to the vcredit_xx.exe

```

```

'     Logging enabled.

```

```

' -----
-----

' 1: Get all the subfolder of the parent folder.

valCounter = 0
set objFolders = objFS0.GetFolder(CurrentDir)
Set arrFolders = objFolders.SubFolders
For Each objFolder in arrFolders
    arrFolderUnsorted(valCounter) = objFolder.Name
    Redim Preserve arrFolderUnsorted(valCounter+1)
    valCounter = valCounter + 1
Next

Redim Preserve arrFolderUnsorted(valCounter-1)
Dim j

' 2: Sort them:

For valCounter = (UBound(arrFolderUnsorted) - 1) to 0 Step
-1
    For j= 0 to valCounter
        If UCase(arrFolderUnsorted(j)) >
UCase(arrFolderUnsorted(j+1)) Then
            strHolder = arrFolderUnsorted(j+1)
            arrFolderUnsorted(j+1) = arrFolderUnsorted(j)
            arrFolderUnsorted(j) = strHolder
        End If
    Next
Next

' 3: Install

For valCounter = 0 to ubound(arrFolderUnsorted) ' I know:
the folders are sorted now....

    if instr(arrFolderUnsorted(valCounter),"2005") > 0 Then
InstallCPlusPlus2005Runtimes
        if instr(arrFolderUnsorted(valCounter),"2008") > 0 Then
InstallCPlusPlus2008Runtimes
            if instr(arrFolderUnsorted(valCounter),"2010") > 0 Then

```

```
InstallCPlusPlus2010Runtimes
    if instr(arrFolderUnsorted(valCounter),"2012") > 0 Then
InstallCPlusPlus2012Runtimes
    if instr(arrFolderUnsorted(valCounter),"2013") > 0 Then
InstallCPlusPlus2013OrLater
    if instr(arrFolderUnsorted(valCounter),"2015") > 0 Then
InstallCPlusPlus2013OrLater
    if instr(arrFolderUnsorted(valCounter),"2017") > 0 Then
InstallCPlusPlus2013OrLater
```

```
next
```

```
        WriteToLog("##### End Installation
#####")
```

```
        WriteToLog("##### Summary of installed C++
Redistributables #####")
```

```
        WriteToLog(" ")
```

```
        Set colSoftware = objWMIService.ExecQuery("Select * from
Win32_Product Where Name LIKE '%Visual C++%' AND (Name LIKE
'%Redistributable%' OR Name LIKE '%Runtime%')")
```

```
        For Each objSoftware in colSoftware
```

```
            WriteToLog("Name: " & objSoftware.Name)
```

```
            WriteToLog("Version: " & objSoftware.Version)
```

```
            WriteToLog(" ")
```

```
        Next
```

```
        WriteToLog(" ")
```

```
        WriteToLog("##### End summary of installed C++
Redistributables #####")
```

```
        WriteToLog(" ")
```

```
        CloseLogFile()
```

```
        wscript.quit 0
```

```
Sub OpenLogFile()
```

```
' -----
-----
```

```
' Subroutine: OpenLogFile()
```

```

' The name of the logfile is mentinoed in the variabele
strOutputFile.
' -----
-----

    If objLogFileFS0.FileExists(strOutputFile) Then
                                Set      objLogFile      =
objLogFileFS0.OpenTextFile(strOutputFile, ForWriting)
    Else
                                Set      objLogFile      =
objLogFileFS0.CreateTextFile(strOutputFile)
    End If

End Sub

Sub CloseLogFile()

' -----
-----
' Subroutine: CloseLogFile()
' Close the log file.
' -----
-----

    WriteToLog "- ACTION: script ended."
    objLogFile.Close
    Set objLogfileFS0 = Nothing

End Sub

Function WriteToLog(sLogMessage)

' -----
-----
' Function: WriteToLog(sLogMessage)
' Writes an entry 'sLogMessage' in the logfile.
' -----
-----

    if instr(sLogMessage, "- ACTION: ") = 0 then sLogMessage = "
" & sLogMessage

```

```

objLogFile.WriteLine("Time: " & now & " " & sLogMessage)

End Function

Sub CreateFolderStructure(strFolderNameToBeCreated)

' -----
' -----
' Subroutine: CreateFolderStructure(strFolderNameToBeCreated)
' Creates the map as mentioned in strFolderNameToBeCreated.
' -----
' -----

    Dim arrFoldersTMP : arrFoldersTMP = split
(strFolderNameToBeCreated, "\")
    Dim strFolder : strFolder = ""
    Dim objFolderTMP

    For Each objFolderTMP in arrFoldersTMP
        strFolder = strFolder & objFolderTMP
        If NOT objFSO.FolderExists(strFolder) Then
            objFSO.CreateFolder(strFolder)
        end If
        strFolder = strFolder & "\"
    Next

End Sub

Function fnKillProcess(strProcessName)

' -----
' -----
' Function: fnKillProcess(strProcessName)
' Terminates the given processname.
' -----
' -----

    Set colProcess = fn_objWMIService.ExecQuery ("Select * From
Win32_Process")
    For Each objProcess In colProcess
        If Instr(LCase(objProcess.Name), LCase(strProcessName)) > 0

```

Then

```
    objShell.Run "TASKKILL /F /T /IM " & objProcess.Name,
0, False
    objProcess.Terminate()
        WriteToLog("Terminating application: " &
objProcess.Name)
    End If
Next
```

End Function

Sub InstallCplusplus2005Runtimes

```
' -----
' -----
' Subroutine: InstallCplusplus2005Runtimes
' Installs the C++ 2005 Runtime modules.
' -----
' -----
```

```
    for Each objVersion in arrVersions
        if instr(arrFolderUnsorted(valCounter),"MFC") > 0 Then
            strLogFile =
right(arrFolderUnsorted(valCounter),len(arrFolderUnsorted(valC
ounter))-3) & " (" & objVersion & ").log"
            strCommand = chr(34) & CurrentDir & "\" &
arrFolderUnsorted(valCounter) & "\vcredist_" & objVersion &
".exe" & chr(34) & " /t:c:\temp /q"
            strCommand = strCommand & ":a /c:" & chr(34) &
"msiexec /i vcredist.msi /qb! /l*v "
            strCommand = strCommand & chr(34) & chr(34) &
strLogLocation & "\" & strLogFile & chr(34) & chr(34) &
chr(34)
            WriteToLog("Running command: " & strCommand)
            valResult = objShell.Run(strCommand, 6, True)
            WriteToLog("Result: " & valResult)
        else
            strCommand = chr(34) & CurrentDir & "\" &
arrFolderUnsorted(valCounter) & "\vcredist_" & objVersion &
".exe" & chr(34) & " /q"
            WriteToLog("Running command: " & strCommand)
            valResult = objShell.Run(strCommand, 6, True)
```

```

        WriteToLog("Result: " & valResult)
    end if
next

End sub

Sub InstallCplusplus2008Runtimes
' -----
' Subroutine: InstallCplusplus2008Runtimes
' Installs the C++ 2008 Runtime modules.
' -----
' -----

    for Each objVersion in arrVersions
                                                strLogFile      =
right(arrFolderUnsorted(valCounter),len(arrFolderUnsorted(valC
ounter))-3) & " (" & objVersion & ").log"
        strCommand = chr(34) & CurrentDir & "\" &
arrFolderUnsorted(valCounter) & "\vcredist_" & objVersion &
".exe" & chr(34) & " /q /l "
        strCommand = strCommand & chr(34) & strLogLocation & "\"
& strLogFile & chr(34)
        WriteToLog("Running command: " & strCommand)
        valResult = objShell.Run(strCommand, 6, True)
        WriteToLog("Result: " & valResult)
    next
End sub

Sub InstallCplusplus2010Runtimes
' -----
' -----
' Subroutine: InstallCplusplus2010Runtimes
' Installs the C++ 2010 Runtime modules.
' -----
' -----

    for Each objVersion in arrVersions
                                                strLogFile      =
right(arrFolderUnsorted(valCounter),len(arrFolderUnsorted(valC
ounter))-3) & " (" & objVersion & ").log"

```

```

        strCommand = chr(34) & CurrentDir & "\" &
arrFolderUnsorted(valCounter) & "\vcredist_" & objVersion &
".exe" & chr(34) & " /passive /norestart /log "
        strCommand = strCommand & chr(34) & strLogLocation & "\"
& strLogFile & chr(34)
        WriteToLog("Running command: " & strCommand)
        valResult = objShell.Run(strCommand, 6, True)
        WriteToLog("Result: " & valResult)
    next

```

End sub

Sub InstallCplusplus2012Runtimes

```

' -----
' -----
' Subroutine: InstallCplusplus2012Runtimes
' Installs the C++ 2012 Runtime modules.
' -----
' -----

```

```

    for Each objVersion in arrVersions
        strLogFile =
right(arrFolderUnsorted(valCounter), len(arrFolderUnsorted(valC
ounter))-3) & " (" & objVersion & ").log"
        strCommand = chr(34) & CurrentDir & "\" &
arrFolderUnsorted(valCounter) & "\vcredist_" & objVersion &
".exe" & chr(34) & " /install /passive /norestart /log "
        strCommand = strCommand & chr(34) & strLogLocation & "\"
& strLogFile & chr(34)
        WriteToLog("Running command: " & strCommand)
        valResult = objShell.Run(strCommand, 6, True)
        WriteToLog("Result: " & valResult)
    next

```

End sub

Sub InstallCplusplus2013orLater

```

' -----
' -----
' Subroutine: InstallCplusplus2013orLater
' Installs the C++ 2013 Runtime modules.

```

```
' -----  
-----  
  
    for Each objVersion in arrVersions  
                                                strLogFile      =  
right(arrFolderUnsorted(valCounter),len(arrFolderUnsorted(valC  
ounter))-3) & " (" & objVersion & ").log"  
        strCommand = chr(34) & CurrentDir & "\" &  
arrFolderUnsorted(valCounter) & "\vcredist_" & objVersion &  
".exe" & chr(34) & " /install /passive /norestart /log "  
        strCommand = strCommand & chr(34) & strLogLocation & "\"  
& strLogFile & chr(34)  
        WriteToLog("Running command: " & strCommand)  
        valResult = objShell.Run(strCommand, 6, True)  
        WriteToLog("Result: " & valResult)  
    next  
  
End sub
```

Detection Rule

Create a rule that indicates the presence of this application.

Setting Type: Registry

Specify the registry key or value to detect this application.

Hive: HKEY_LOCAL_MACHINE Browse...

Key: Software\Microsoft\VisualStudio\14.0\VC\Runtimes\x64

Value: Version

Use (Default) registry key value for detection

This registry key is associated with a 32-bit application on 64-bit systems

Data Type: String

This registry setting must exist on the target system to indicate presence of this application

This registry setting must satisfy the following rule to indicate the presence of this application

Operator: Equals

Value: v14.16.27024.01

OK Cancel

C++ Detection Rule

- [Microsoft_CPlusPlusRuntimes_1.0.1_ENG_WIN7_P1: Link to the files, including the code.](#)
- [Microsoft_CPlusPlusRuntimes_2.0.0_ENG_WIN7_P1: Link to the files, including the code.](#)