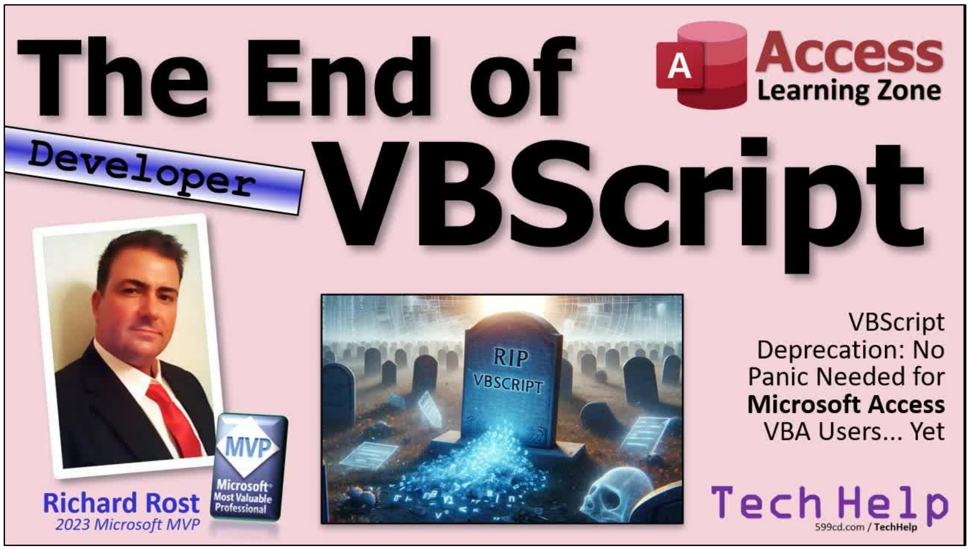
End of VBScript

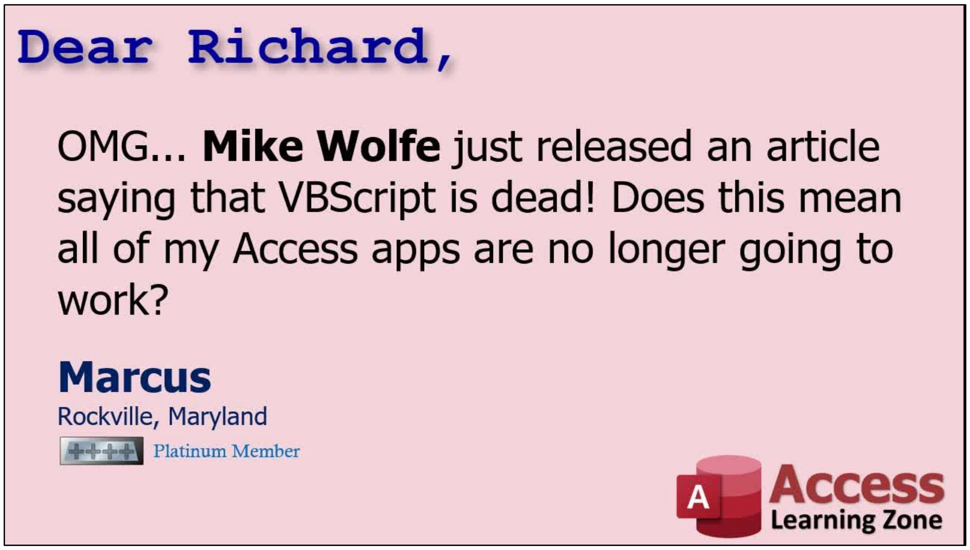


The End of VBScript

Developer

VBScript Deprecation: No Panic Needed for **Microsoft Access** VBA Users... Yet

Welcome to another TechHelp video brought to you by AccessLearningZone.com. I'm your instructor, Richard Rost. I had to address this topic, folks. I've been receiving a substantial amount of email inquiries about it. Today's topic is about the end of VBScript.



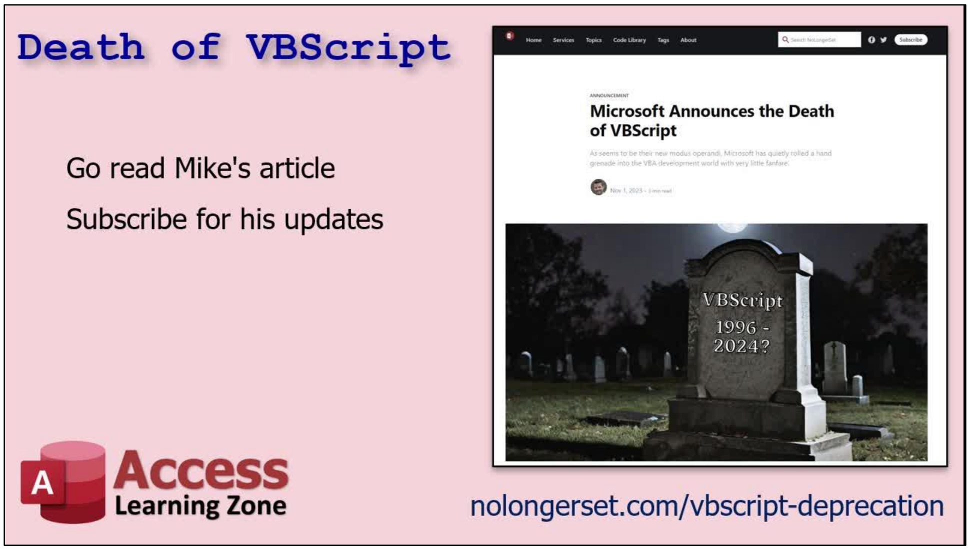
Dear Richard,

OMG... **Mike Wolfe** just released an article saying that VBScript is dead! Does this mean all of my Access apps are no longer going to work?

Marcus
Rockville, Maryland
Platinum Member

Access Learning Zone

What is VBScript, and how does it affect your Microsoft Access databases? We're going to examine that today. This has been a frequent subject in emails over the past week, prompting me to provide my insights.



Death of VBScript

Go read Mike's article

Subscribe for his updates

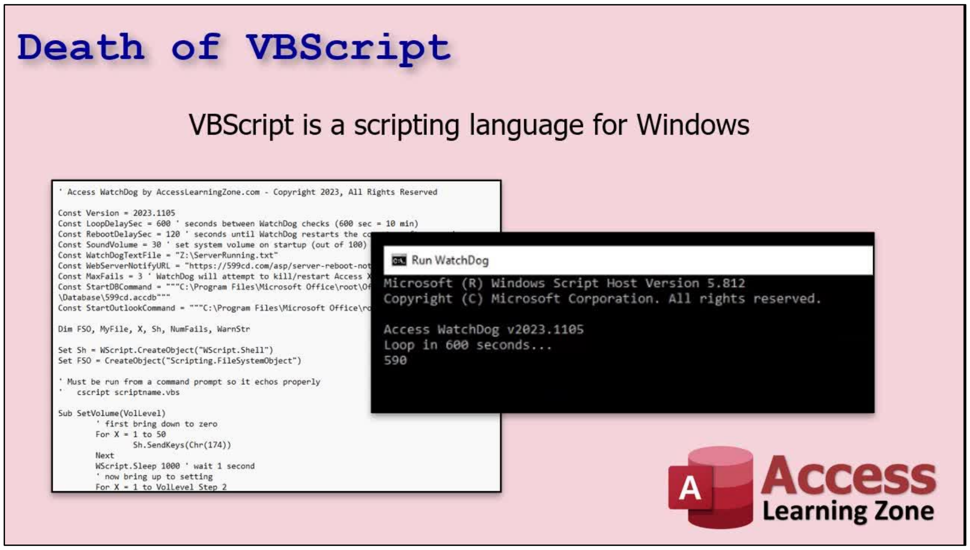
Microsoft Announces the Death of VBScript

As seems to be their new modus operandi, Microsoft has quietly rolled a hand grenade into the VBA development world with very little fanfare.

Nov 1, 2023 · 3 min read

nolongerset.com/vbscript-deprecation

I selected one email to share. Marcus from Rockville, Maryland, one of my Platinum Members, writes, "Oh my God, Microsoft just released an article about a week ago, stating that VBScript is dead. Does this mean that all my Access apps will no longer work?" The short answer is no, not necessarily. But let's delve into the details. If you haven't yet read Mike's article, I recommend you do so; there's a link. He discusses the topic with much more depth than I will, as I am just covering the basics. While you're on his website, make sure to subscribe to his email updates. He publishes outstanding articles and delivers something nearly every day at a discounted rate. Although Mike's content leans more toward written material, and I'm more of a video guy, he closely monitors the latest on Access, more than I do. I usually get my news from him and Access Forever. Whenever I see a new post from him in my inbox, I make it a point to read it.



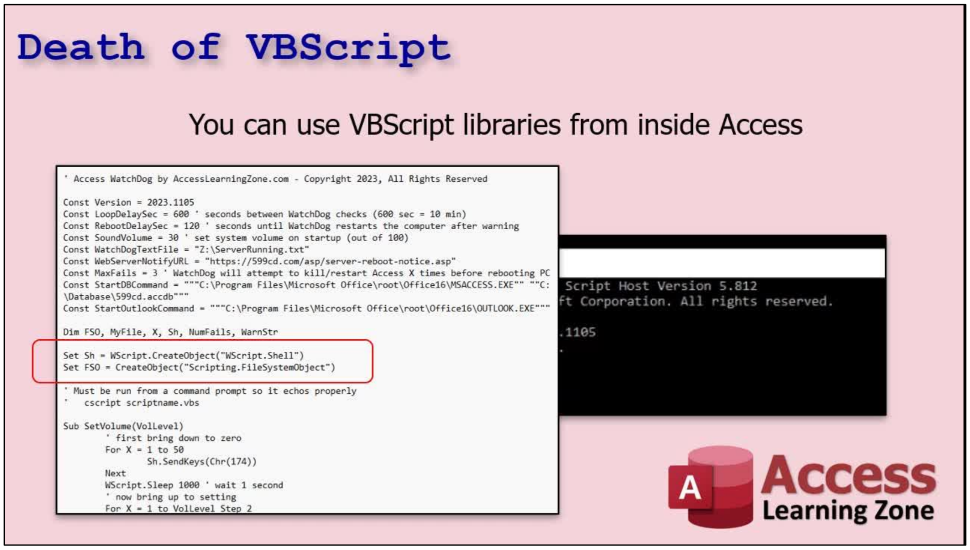
Death of VBScript

VBScript is a scripting language for Windows

```
' Access WatchDog by AccessLearningZone.com - Copyright 2023, All Rights Reserved

Const Version = 2023.1105
Const LoopDelaySec = 600 ' seconds between WatchDog checks (600 sec = 10 min)
Const RebootDelaySec = 120 ' seconds until WatchDog restarts the co
Const SoundVolume = 30 ' set system volume on startup (out of 100)
Const WatchDogTextFile = "Z:\ServerRunning.txt"
Const WebServerNotifyURL = "https://599cd.com/asp/server-reboot-not
Const MaxFails = 3 ' WatchDog will attempt to kill/restart Access X
Const StartDBCommand = """C:\Program Files\Microsoft Office\root\Of
\Database\599cd.accdb"""
Const StartOutlookCommand = """C:\Program Files\Microsoft Office\ro

Dim FSO, MyFile, X, Sh, NumFails, WarnStr

Set Sh = WScript.CreateObject("WScript.Shell")
Set FSO = CreateObject("Scripting.FileSystemObject")

' Must be run from a command prompt so it echos properly
'   cscript scriptname.vbs

Sub SetVolume(VolLevel)
        ' first bring down to zero
        For X = 1 to 50
                Sh.SendKeys(Chr(174))
        Next
        WScript.Sleep 1000 ' wait 1 second
        ' now bring up to setting
        For X = 1 to VolLevel Step 2
```

```
Microsoft (R) Windows Script Host Version 5.812
Copyright (C) Microsoft Corporation. All rights reserved.

Access WatchDog v2023.1105
Loop in 600 seconds...
590
```

Now, for those of you who are unaware, VBScript is essentially a tool that allows you to write VB-like code and run it directly in Windows, similar to the batch files from the days of DOS and Windows ninety-five. I still utilize batch files quite frequently. VBScript can be thought of as an enhanced version of batch files. For instance, I created a utility known as the Access Watchdog. It operates in a loop and monitors your Access database. If the database freezes for any reason, the Watchdog will restart it. This is particularly useful if you're employing Access as a database server, and the entire software was crafted in VBScript.



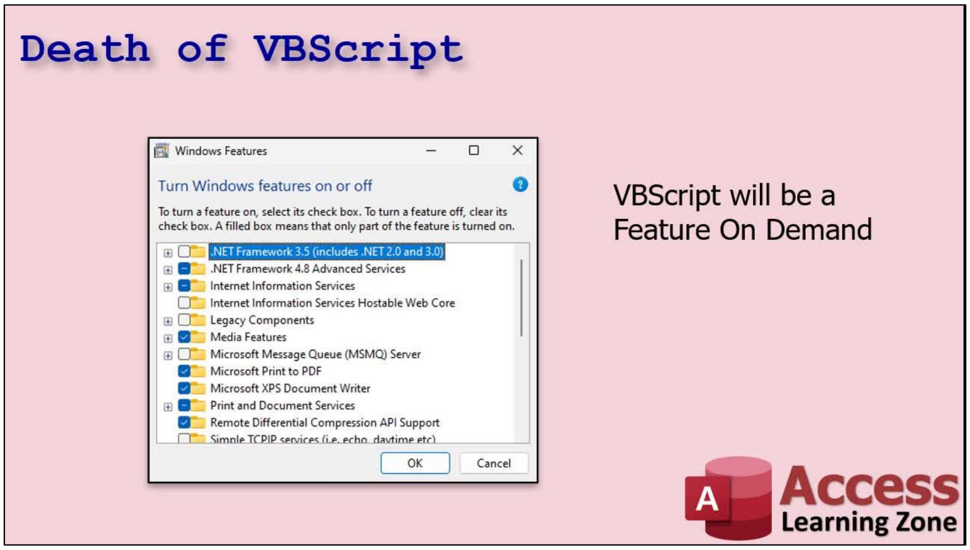
Death of VBScript

You can use VBScript libraries from inside Access

```
' Access WatchDog by AccessLearningZone.com - Copyright 2023, All Rights Reserved

Const Version = 2023.1105
Const LoopDelaySec = 600 ' seconds between WatchDog checks (600 sec = 10 min)
Const RebootDelaySec = 120 ' seconds until WatchDog restarts the computer after warning
Const SoundVolume = 30 ' set system volume on startup (out of 100)
Const WatchDogTextFile = "Z:\ServerRunning.txt"
Const WebServerNotifyURL = "https://599cd.com/asp/server-reboot-notice.asp"
Const MaxFails = 3 ' WatchDog will attempt to kill/restart Access X times before rebooting PC
Const StartDBCommand = """C:\Program Files\Microsoft Office\root\Office16\MSACCESS.EXE"" ""C:
\Database\599cd.accdb"""
Const StartOutlookCommand = """C:\Program Files\Microsoft Office\root\Office16\OUTLOOK.EXE"""

Dim FSO, MyFile, X, Sh, NumFails, WarnStr

Set Sh = WScript.CreateObject("WScript.Shell")
Set FSO = CreateObject("Scripting.FileSystemObject")

' Must be run from a command prompt so it echos properly
'   cscript scriptname.vbs

Sub SetVolume(VolLevel)
        ' first bring down to zero
        For X = 1 to 50
                Sh.SendKeys(Chr(174))
        Next
        WScript.Sleep 1000 ' wait 1 second
        ' now bring up to setting
        For X = 1 to VolLevel Step 2
```

```
 Script Host Version 5.812
ft Corporation. All rights reserved.

.1105
```

The Access Watchdog functions in a simple DOS window, checking the status of the database every 10 minutes. Essentially, VBScript is a streamlined scripting language for Windows. Even if you haven't used VBScript personally, which is fine as I don't use it much myself, it's important to recognize that several developers do utilize VBScript within their Access VBA. For example, in one of my developer lessons, I teach how to employ the file system object to retrieve information on files and folders, such as file sizes and modification dates.



Death of VBScript

Windows Features

Turn Windows features on or off

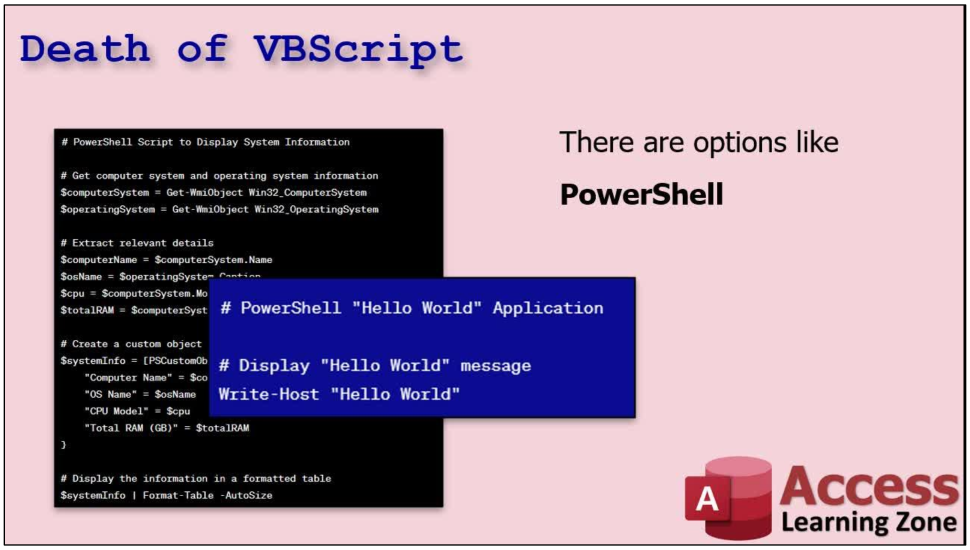
To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.

- ⊞ □ .NET Framework 3.5 (includes .NET 2.0 and 3.0)
- ⊞ ■ .NET Framework 4.8 Advanced Services
- ⊞ ■ Internet Information Services
- □ Internet Information Services Hostable Web Core
- ⊞ □ Legacy Components
- ⊞ ☑ Media Features
- ⊞ □ Microsoft Message Queue (MSMQ) Server
- ☑ Microsoft Print to PDF
- ☑ Microsoft XPS Document Writer
- ⊞ ■ Print and Document Services
- ☑ Remote Differential Compression API Support
- □ Simple TCPIP services (i.e. echo, daytime etc)

VBScript will be a Feature On Demand

While VBScript is losing favor, it's uncertain whether components like the file system object will remain supported. There are alternative methods for handling files in VBA, but the file system object's capabilities are superior. One notable change highlighted in Mike's article is the shift of VBScript from an automatically installed feature in Windows to a 'feature on demand.' This means developers wishing to use VBScript on a device—whether their own, a client's, or wherever the database is hosted—will need to activate it in the 'Turn Windows features on and off' settings, which will include VBScript in the future. This option should be available for some time, anticipated to be a few years, though I cannot say for certain and do not claim insider knowledge at Microsoft. Make sure to stay informed through professionals like Mike and Access Forever as they will likely know before I do. VBScript is not vanishing immediately; it will transition gradually, providing developers time to adapt their applications.

Even though Outlook is eliminating COM objects, which enable inter-Office application communication, it seems improbable that VBA will be phased out soon.



Death of VBScript

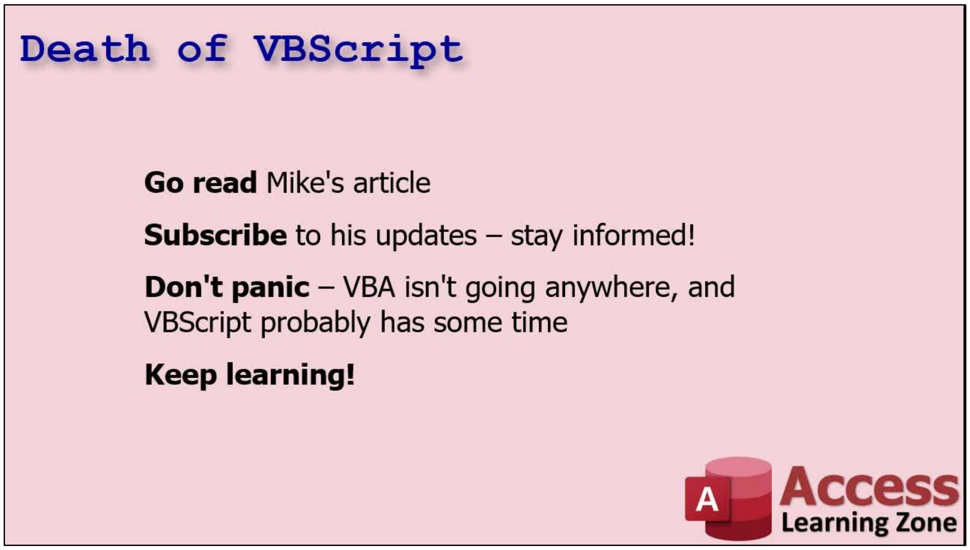
```
# PowerShell Script to Display System Information

# Get computer system and operating system information
$computerSystem = Get-WmiObject Win32_ComputerSystem
$operatingSystem = Get-WmiObject Win32_OperatingSystem

# Extract relevant details
$computerName = $computerSystem.Name
$osName = $operatingSyste
$cpu = $computerSystem.Mo
$totalRAM = $computerSyst

# Create a custom object
$systemInfo = [PSCustomOb
    "Computer Name" = $co
    "OS Name" = $osName
    "CPU Model" = $cpu
    "Total RAM (GB)" = $totalRAM
}

# Display the information in a formatted table
$systemInfo | Format-Table -AutoSize
```

```
# PowerShell "Hello World" Application

# Display "Hello World" message
Write-Host "Hello World"
```

There are options like

PowerShell

Microsoft has a vast network of developers who have created a multitude of applications using VBA for decades across Access, Excel, Word, PowerPoint, and more. A sudden eradication of VBA could incite a mutiny among users. And, from my perspective, it's unlikely that VBA will be discontinued in the near term. Although I am fond of VBA, I remember my nostalgia for Color Basic on my TRS-80; preferences can change over time. Fortunately, there are various programming language alternatives, such as C++, VB.NET, and others.



Death of VBScript

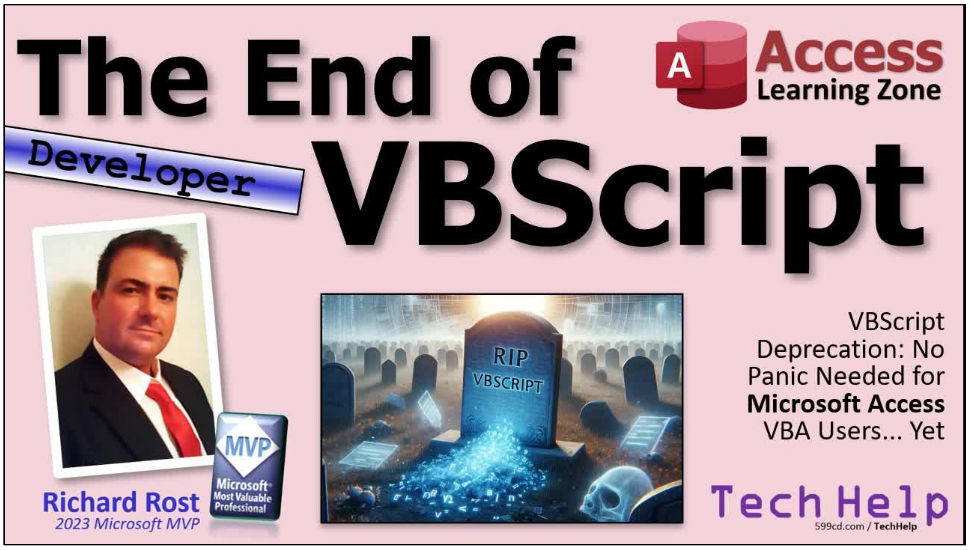
Go read Mike's article

Subscribe to his updates – stay informed!

Don't panic – VBA isn't going anywhere, and VBScript probably has some time

Keep learning!

For those interested in scripting languages, PowerShell is worth exploring. I've been studying PowerShell myself and, once proficient, may develop courses on it. As alternatives always exist, the key is not to panic. AI tools like ChatGPT can aid in scripting tasks. For example, as you can see, I asked for a Hello World script in PowerShell, and ChatGPT delivered. I've employed AI to help convert VBScript code to PowerShell, but always verify the code works since AI can produce unusual results—this includes OpenAI and GPT-generated content.



Be sure to read Mike's article, sign up for his updates, and stay informed with resources like Access Forever—tell Carl I sent you. Remember, there's no need to panic. VBA remains intact for the foreseeable future, and VBScript will likely remain accessible as a feature on demand for some time. It's crucial to embrace continual learning; I strive to learn new things daily and hope that your exploration of these topics is enriching. Here is my condensed perspective on the matter. Find further information and links beneath this video. That concludes today's TechHelp video. I hope you've gained some knowledge and remain calm. Live long and prosper, friends. I'll see you next time.