Using the Jet ShowPlan Manager – Version 2.4

1. Introduction

The **Jet ShowPlan** feature is used to view the execution plan of Access queries and SQL statements.

The **query execution plan** is a set of instructions to the database engine that tell it how to execute a query.
As a simple example, consider a query that retrieves all customers located in the UK.
One way to do this would be to examine every record and select the ones where the Country field equals UK.
But if there's an index on that field, a more efficient way to perform the same query would probably be to examine the index, and then jump straight to the records from UK.

The following information is taken from an excellent article by **Susan Haskins** written in 2003:
https://www.techrepublic.com/article/use-microsoft-jets-showplan-to-write-more-efficient-queries/

Jet creates this plan each time you compile the query – for example the first time you run it, when you save a change to the query, or when you compact the database. Jet uses this plan behind the scenes to determine the quickest way to go about executing the query. Once the plan exists, Jet simply refers to the plan to run the query instead of re-evaluating the query each time you run it. One easy way to optimize a query is to compact the database if you make several changes to the data or add a lot of new data. Doing so will force a re-evaluation of the query plan. What works best for ten rows might not be the best plan for 10,000 records. The plan contains information on the following components:

- *WHERE or HAVING clauses*
- *ORDER BY clause*
- *Joins*
- *Indexes*
- *Table stats*

Additional information bringing this article up to date is contained in another article ***Show Plan – Run Faster*** on my website : http://www.mendipdatasystems.co.uk/show-plan-go-faster/4594460516

In order to use the **JET ShowPlan** feature, you first need to setup the feature in the registry. To do this requires knowledge of the correct locations for several registry keys, some of which are version dependant.

The **JET ShowPlan Manager** application is designed to make this process as simple as possible

2. Using the application

In order to setup the **JET ShowPlan** feature, **Access MUST be run as an administrator.**

To do so, **right click** on the Access shortcut in the start menu or desktop and click **Run As Administrator.**
If this option isn't available (e.g. Access 2010), **hold the shift key down** as you right click the shortcut

Access versions prior to 2007 cannot be run as an administrator so no MDB version is available for this utility

When the application first opens, it will collect information about the version of Windows and access being used. It will also determine whether these are 32-bit or 64-bit and whether a copy of Office 365 is installed.

This information is needed to determine the correct registry path needed for the JETSHOWPLAN string value

This process will take a few seconds and the result will look similar to one of the screenshots below:

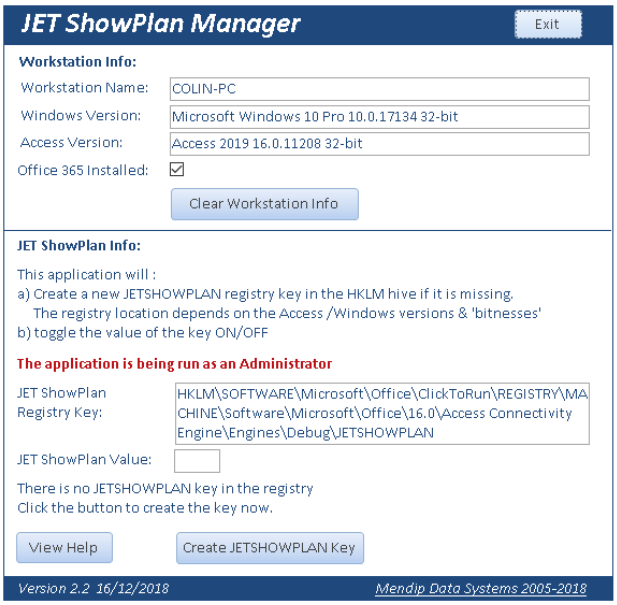


Figure 1

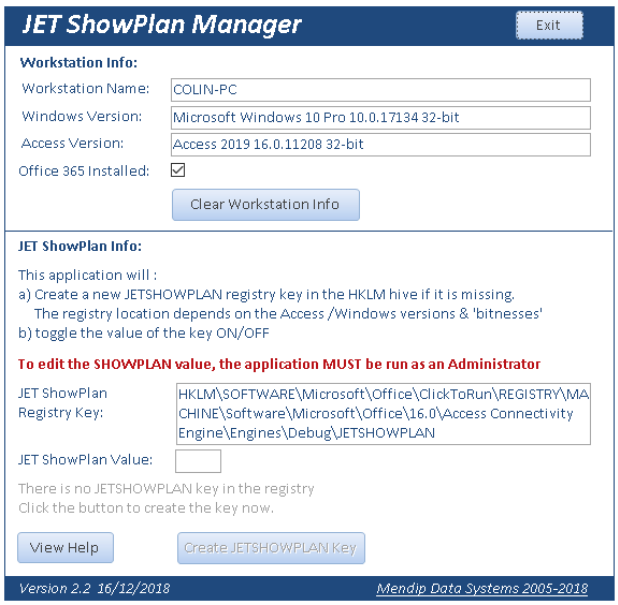


Figure 2

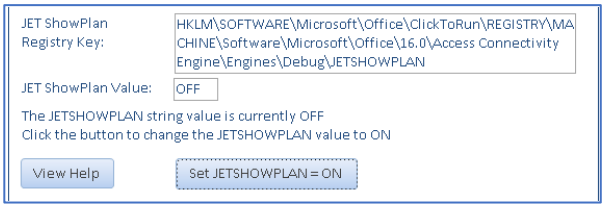
If the application was **not being run as an administrator**, the screen would look like ***Figure 2*** with parts of the screen **disabled**. If so, **close and reopen** using the **run as administrator** option

In the examples shown above, a 32-bit version of Access 365 is being run on 32-bit Windows

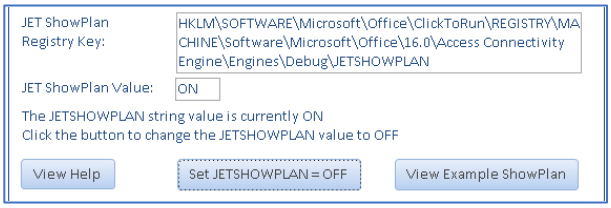
The correct registry key in this case for the JETSHOWPLAN string is:
HKLM\SOFTWARE\Microsoft\Office\ClickToRun\REGISTRY\MACHINE\Software\Microsoft\Office\16.0\Access Connectivity Engine\Engines\Debug\JETSHOWPLAN

The registry path depends on the Windows 'bit-ness', the Access version and bit-ness and whether or not it is an Access 365 installation. For further details on the various paths, see section 3: How the application works

The registry key is not created automatically when Access is installed.
Click the ***Create JETSHOWPLAN Key*** button to do so.
After a couple of seconds, the screen will be updated with the **JETSHOWPLAN** key value is set to OFF

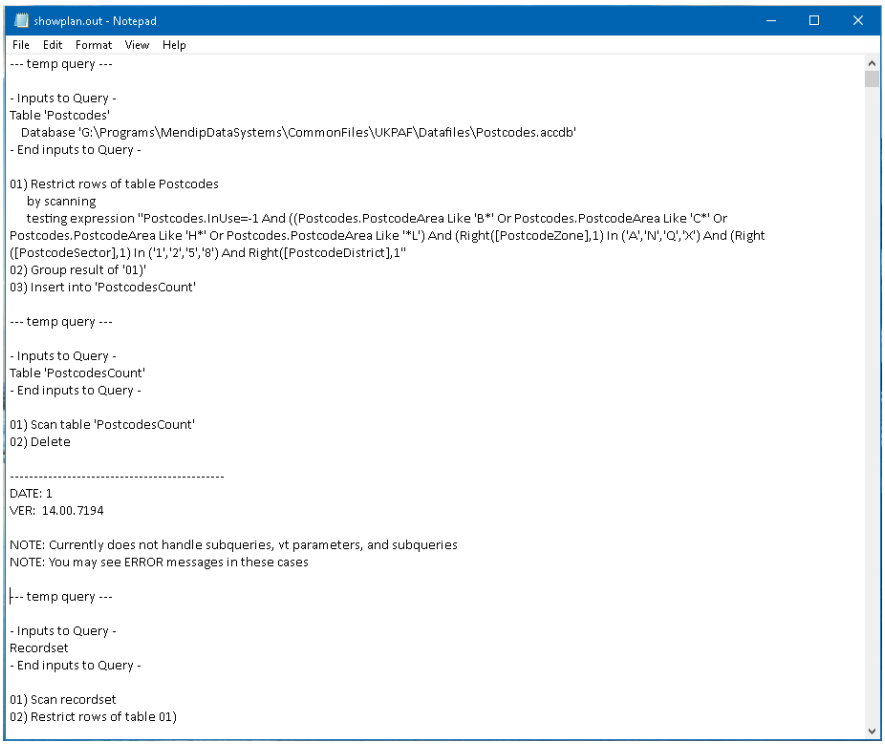


The button caption has changed to ***Set JETSHOWPLAN = ON.***
Click the button again to **enable** the feature.



Whilst the JETSHOWPLAN value = ON, the execution plan of every query or SQL statement used by this version of Access will be saved to a plain text file ***showplan.out*** in the **default database directory.**

For example:



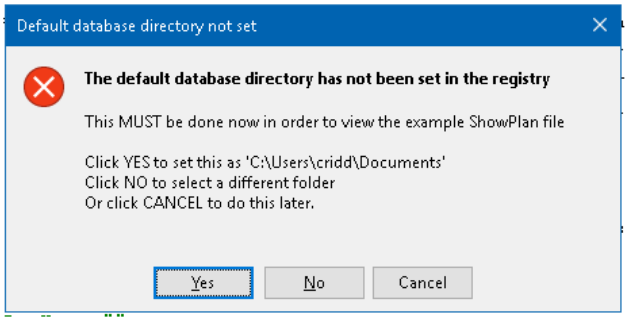
The same file is used each time so it can over time become very large indeed.
In addition, the time needed to complete queries increases by around 14% when the feature is ON
It is therefore strongly recommended that JETSHOWPLAN is switched OFF when it isn't required.

To view an example ***showplan.out*** file, click the **View Example ShowPlan** button.
This will check the **default database directory** in the **registry**, associate **.out files** with **Notepad**, run a simple query qryComputerInfo and then open the ***showplan.out*** file in **Notepad**

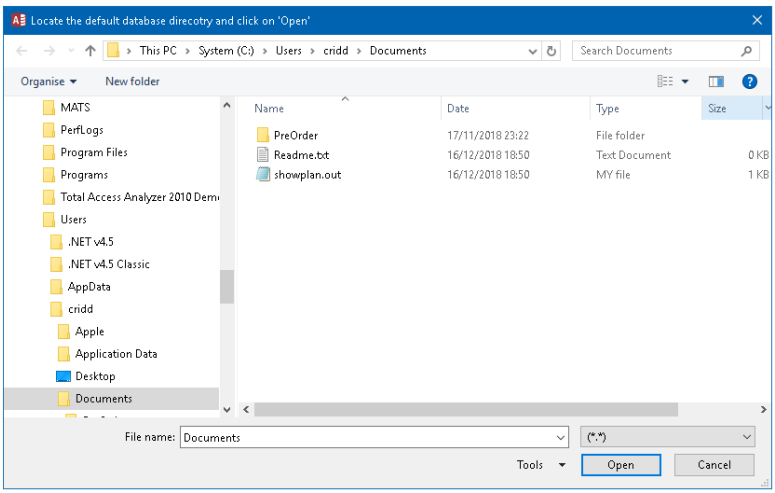
When Access is installed, the default database directory is usually set as C:\Users***UserName***\Documents
OR if you are using a Microsoft account it may be set as C:\Users***UserName******OneDrive***\Documents

NOTE:
It is STRONGLY recommended that OneDrive is NOT used as the default directory. As that requires an online connection, any interruptions to that connection can cause corruption leading to loss of data and/or an unusable database

Unfortunately, the default database directory is only stored in the registry if it is changed!
In order to view the show plan file, the application will next add the default database directory key & value to the registry if it doesn't already exist. If so, a message box similar to this will appear:



Clicking **YES**, will set the default to your ***My Documents*** area e.g. C:\Users\cridd\Documents.
If that location isn't correct, click **NO** instead. A **Browse** folder dialog will appear.



Browse to and select the folder required then click OK.
The new default folder will be implemented next time Access is opened.
If you have **changed the default folder**, this will affect **all new databases created** from now on.

Restart the application – remember to use **Run As Administrator**

Click the **View Example ShowPlan** button again. This will run a simple query and open the ***showplan.out*** file which was saved in your default database directory.

NOTE: If you still have **OneDrive** as your default directory, the ***showplan.out*** file will be created and may flash briefly but then close. Another good reason not to use this online folder area!

You can now use the **JET ShowPlan** feature to assist with optimising queries and SQL statements
Do remember to switch this feature **OFF** in the registry when not required for query optimisation

3. How the Application Works

a) Various functions (in the module ***modSysInfo***) are used to identify the Windows & Access versions/bitnesses:
   - Windows – ***GetOperatingSystem / IsWin32OrWin64***
   - Access – ***GetAccessVersion / GetAccessEXEVersion / IsOfficex64***

   The results are stored in the table **tblComputerInfo**

   If the application is reopened on a new workstation or using a different version of Access, this data is automatically deleted and the table repopulated.

b) Checking whether **Office 365** is installed is more difficult.
   However, it is necessary to do so in order to use the **Jet ShowPlan** feature.
   As Microsoft uses the same version numbers for both retail Office and the Office 365 subscription model, the approach used is to check the registry.
   The code to do this is in the ***CheckAccess365*** function (in ***modFunctions***).

   It is partly because this function needs to read from the *HKEY_LOCAL_MACHINE (HKLM)* registry hive that the application MUST be run as an administrator.

   All retail versions of Access from 2007 onwards include a registry key similar to:
   *HKEY_LOCAL_MACHINE\ SOFTWARE\Microsoft\Office**16.0**\Access Connectivity Engine\Engines*
   The part in **bold** is the Access version number

   For 32-bit Access on 64-bit Windows, this changes to:
   *HKEY_LOCAL_MACHINE\ SOFTWARE**Wow6432Node**\Microsoft\Office**16.0**\Access Connectivity Engine\Engines*

   In each case, the Engines key contains a string ***SystemDB*** with value ***system.mdb***
   However, for Office 365 installations, a **ClickToRun** registry structure is used instead.

   For 32-bit Windows, only 32-bit Access can be installed. The function will check for the existence of the registry key:
   *HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office**ClickToRun\REGISTRY\MACHINE\Software\Microsoft\Office**16.0\Access Connectivity Engine\Engines*

   For 64-bit Access in 64-bit Windows, the same registry key is used as above

   However, for 32-bit Access 365 on 64-bit Windows, this changes to:
   *HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office**ClickToRun\REGISTRY\MACHINE\Software\Wow6432Node\Microsoft\Office**16.0\Access Connectivity Engine\Engines*

   So, if one of the above keys is found, Office 365 is installed.

   However, it is not that 'simple'. If a **retail version** of **Office 2013/2016/2019** is installed but the user enters their **Microsoft account** information either during installation or at a later time, this triggers the **ClickToRun** registry structure to be created!

   In other words, it is then treated as Office 365 even though it is still a retail product.
   However, the software is not updated with new features as is the case with a true Office 365 product

Furthermore, in such cases, a slightly different registry structure MAY also be created such as:
*HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office**15.0**\ClickToRun\REGISTRY\MACHINE\Software\Microsoft\Office\15.0\Access Connectivity Engine\Engines*
or
*HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office**16.0**\ClickToRun\REGISTRY\MACHINE\Software**Wow6432Node**\Microsoft\Office\16.0\Access Connectivity Engine\Engines*

As a result, the code looks for any of those key structures to determine whether Office 365 is installed.
Why Microsoft decided to make this so very difficult is hard to understand!

c) Once the correct registry key for the **SystemDB** string value has been determined, this is stored in the table **tblComputerInfo**. It is then used to create a new **Debug** key in the Engines 'folder' with a string value **JETSHOWPLAN** and a data value ***OFF*** or ***ON***.

d) The **default database directory** string value is stored in the registry folder similar to:
*HKEY_CURRENT_USER\SOFTWARE\Microsoft\Office**14.0**\Access\Settings*
The part in **bold** is the Access version number
The path is NOT dependant on whether Access is a retail version or a 365 subscription

e) The default application used for the ***showplan.out*** file type is not specified automatically.
As it is a plain text file, the application associates the **.out** file type with **Notepad.**
It does this by writing new keys in these sections of the registry hives HKCR & HKCU:
- *HKEY_CLASSES_ROOT\.out*
- *HKEY_CLASSES_ROOT\. OUT file*
- *HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\FileExts\.out\Application*
- *HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\FileExts\.out\OpenWithList*

I have explained the registry paths in some detail as the same ideas could, in principle, be used to modify other features of Access using appropriate registry keys.

4. Reading the registry Wow6432Node

When using 32-bit applications in 64-bit Windows, registry entries are made to the **Wow6432Node.**
Windows uses a process called **registry redirection** to manage this area but that causes issues when trying to view those entries from a 32-bit application such as Access
This area cannot be read from or written to using standard code as used on 'pure' 32-bit or 64-bit systems.
Instead, I have used 'special' code which is elevated to 64-bit as and when required

```
Public Function GetStdRegProv() As Object
' http://msdn.microsoft.com/en-us/library/aa394600(VS.85).aspx
'code updated by Jeff Holm to manage mixed mode systems

On Error GoTo ErrHandler:

   Dim strComputer As String

   strComputer = "."

   If IsWin32OrWin64 = IsOfficex64 Then    'Office & Windows bitness match, no need to force 64-bit

      Set GetStdRegProv = GetObject("winmgmts:" _
                        & "{impersonationLevel=impersonate}!\\" _
                        & strComputer & "\root\default:StdRegProv")

   Else    'Office 32-bit & Windows 64-bit, so have to elevate GetStdRegProv to 64-bit

      Dim objCtx As Object
      Dim objLocator As Object
      Dim objServices As Object
      Dim objStdRegProv As Object

      Set objCtx = CreateObject("WbemScripting.SWbemNamedValueSet")
      objCtx.Add "__ProviderArchitecture", 64
      objCtx.Add "__RequiredArchitecture", True
      Set objLocator = CreateObject("Wbemscripting.SWbemLocator")
      Set objServices = objLocator.ConnectServer(strComputer, "root\default", "", "", , , , objCtx)
      Set GetStdRegProv = objServices.Get("StdRegProv")

      Set objServices = Nothing
      Set objLocator = Nothing
      Set objCtx = Nothing
   End If

Exit_ErrHandler:
   On Error Resume Next
   Exit Function

ErrHandler:
   If Err.Number >= 0 Then
      MsgBox "Error " & Err.Number & " in GetStdRegProv procedure: " & Err.description, vbOKOnly +
vbCritical
   End If
   Resume Exit_ErrHandler

End Function
```

5. Acknowledgments

I am extremely grateful to Utter Access forum member **Jeff Holm** for repeatedly testing different versions of this application in mixed **32/64 bit** systems. Also for making several valuable suggestions and providing code snippets used for solving issues with registry keys using the **Wow6432Node** without having to deal with the complexities of ***registry redirection***.

Also, thanks are due to Tom Stiphout, Dev Ashish and Daniel Pineault for various items of standard code used in this application'

I would be grateful for any feedback related to this application.
To do so, please email me at: info@mendipdatasystems.co.uk.
Alternatively, use the contact page on my website: http://www.mendipdatasystems.co.uk

Colin Riddington *Mendip Data Systems*