

Use Microsoft Jet's ShowPlan to write more efficient queries

In large databases, an inefficient query can reduce performance to a crawl and incur the ire of users. The ShowPlan feature of the Access Jet engine can help you plan more efficient queries, optimize your database, and increase performance.

By [Susan Harkins](#) | August 28, 2003, 12:00 AM PST

By Susan Sales Harkins and Mike Gunderloy

With today's speedy systems, performance can take a back seat during the development stages, because most of the tasks return such quick results. The problem is you might not realize that the current design is inefficient until things slow down and your client or users begin to call and complain. Where queries are concerned, you can find help with the ShowPlan feature of Microsoft's Jet engine—an option that prints the query's plan so you can review and then improve upon the design if possible. In this article, we'll show you how to use ShowPlan to optimize your queries.

About query optimization

Regardless of how you state your query, Jet will run that query using the most efficient plan. In fact, if you use the query design grid, Access sometimes rearranges criteria expressions and references when you switch from Datasheet View back to the query design window. That's Jet's query optimization at work. Access rearranges things because your way isn't the most efficient way to run the query. You don't need to worry about these changes, because your query will return exactly the same results, it will just do so quicker.

Behind the scenes, a query has another version. The query plan is a set of instructions to the Jet engine that tell it how to execute a query. For a simple example, consider a query that retrieves all customers located in Alaska. One way to do this would be to examine every record and pull out the ones where the State field equals Alaska. 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 Alaska.

Jet creates this plan each time you compile the query (e.g., 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

About ShowPlan

The ShowPlan option was added to Jet 3.0, and produces a text file that contains the query's plan. (ShowPlan doesn't support subqueries.) You must enable it by adding a Debug key to the registry like so: `\\HKEY_LOCAL_MACHINE\SOFTWARE\MICROSOFTJET\4.0\Engines\Debug`

Under the new Debug key, add a string data type named JETSHOWPLAN (you must use all uppercase letters). Then, add the key value ON to enable the feature. If Access has been running in the background, you must close it and relaunch it for the function to work.

When ShowPlan is enabled, Jet creates a text file named SHOWPLAN.OUT (which might end up in your My Documents folder or the current default folder, depending on the version of Jet you're using) every time Jet compiles a query. You can then view this text file for clues to how Jet is running your queries. We recommend that you disable this feature by changing the key's value to OFF unless you're specifically

using it. Jet appends the plan to an existing file and eventually, the process actually slows things down. Turn on the feature only when you need to review a specific query plan. Open the database, run the query, and then disable the feature.

Creating the new key

Actually adding a key to the registry and specifying the new keys values isn't exactly intuitive, so in this section, I'll show you how to add this new key, step-by-step. Before doing so, you should know that the registry can be a dangerous tool if you don't know what you're doing. Always create a backup of the registry before you attempt to modify it. That way, if anything goes wrong, you can revert to your preproblem settings. Some developers may cringe at the thought of the uneducated traipsing around in their registry, but if you're careful you should have no problems.

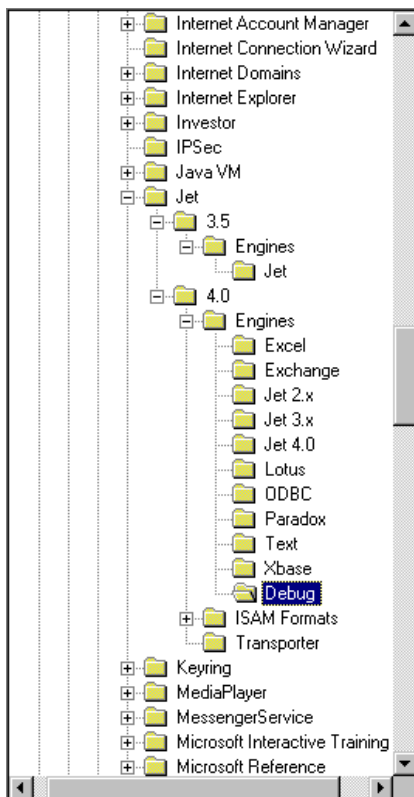
The first step is to actually open the registry. Click the Windows Start button on the Taskbar, choose Run, and enter *regedit*. Click OK to open the Registry Editor. This editor allows you to view and change registry settings—the registry controls how your computer runs. Specifically, Windows stores configuration information in a database (i.e., the registry), which the editor then displays in a tree format so you can more easily find, access, and modify those settings.

Use the tree view in the left pane to locate the appropriate key, which in this case is:

\\KEY_LOCAL_MACHINE\SOFTWARE\MICROSOFT\JET\4.0\Engines

Select the Engines key. You may have more than one engine version on your system; you'll be altering the 4.0 key. With the Engines key selected, choose New from the Edit menu and then select Key. The editor will display a new key node in Edit mode. While in Edit mode, enter the name of the key—Debug, as shown in **Figure A**.

Figure A

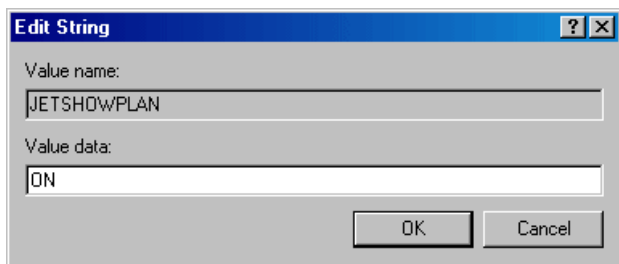


Name the new key.

To set the new key's value, with Debug still selected in the left pane, select New from the Edit menu. Select String Value from the resulting submenu. The editor will display a new value in the right pane, in

Edit mode. Enter *JETSHOWPLAN* (all in uppercase letters) as the value's name. Then, right-click this item, choose Modify, and enter *ON* in the Value Data control, as shown in **Figure B**.

Figure B



Enable the new SHOWPLAN key by entering the value ON.

Click OK to return to the editor and the completed key shown in **Figure C**. At this point you can close the Registry. If Access is open, close it and relaunch it.

Figure C

Name	Type	Data
(Default)	REG_SZ	(value not set)
JETSHOWPLAN	REG_SZ	ON

The completed key is enabled.

Using SHOWPLAN.OUT

Now it's time to run a query or two and view the results in SHOWPLAN.OUT—the text file I mentioned. First, open Northwind, the sample database that comes with Access. Then, you'll need to run the query from the SQL window. To do so, click the Queries shortcut in the Database window, and then double-click the Create Query In Design View shortcut in the object list. When Access opens the blank design grid, close the Show Table dialog box, and then click the View button, which will default to SQL View. Enter (or cut and paste) the SQL statement shown in [Listing A](#) into the SQL window and then run it.

Using a text editor (Microsoft Word for example), locate SHOWPLAN.OUT and open it to see the results as shown in [Listing B](#).

The plan starts by showing the information that Jet has available to help it evaluate your query expression. As you can see, it depends primarily on indexes, and it's chosen appropriate indexes for each table in this query. The numbered steps then show how Jet intends to get the answer you're looking for.

Notice that the first step reveals that a table scan is being run against Orders to find each ShipCity value that equals Madrid. A table scan is generally considered inefficient because Jet reviews each record in the table. There are hundreds of records in the Orders table, but very few of them actually go to Madrid. To improve performance, simply assign a simple index to ShipCity in the Orders table. After creating the index, save the table and close it.

Reopen the query in Design View and save it again to generate a new plan and run the query. Now, open SHOWPLAN.OUT again to reveal [Listing C](#). Remember, SHOWPLAN.OUT is cumulative, meaning it will append the new plan to the existing contents of the file. Scroll to the end of the text file to see the most recent entry.

The restriction on Orders is now performed with Rushmore rather than a table scan, and Rushmore is

definitely more efficient. Rushmore is a query optimization technique that Access inherited when Microsoft bought the FoxPro database. With Rushmore, Jet can determine which records should be returned from a table by reading only the index, rather than reading the actual data. This offers a great speed increase in many cases, because indexes are smaller than data and so require less disk access and network bandwidth to read.

Admittedly, this one change may not seem like a big deal, and similar changes won't always improve performance noticeably. If you're dealing with a small amount of data, chances are you won't notice any change. The real potential is realized when you're dealing with huge tables with thousands of records that continually increase as more data is added. Under these circumstances, a simple switch from a table scan to an index using Rushmore can produce a substantial improvement in performance. In short, if the plan shows restriction by scanning, you've got a good candidate for indexing.

Another thing you should check your query plans for is whether Jet is using the indexes that you think it should. Sometimes reading the first part of the plan will show you an error in your database design. For example, if you've created an index that jointly covers the city and state fields in a table, Jet won't use it to resolve queries that only search on the city. In that case, you might be better off with separate indexes rather than a joint one.

Plan it right

As a competent database developer, you want your applications to run as quickly and smoothly as possible. Don't wait for data to slow down your custom applications—use the most optimized plan possible right from the start.