

by Peter Martin

RPG and Cobol developers spend a lot of time with Display Database Relations (DSPDBR) and Display File Description (DSPFD) trying to work out which access path is the most appropriate to use. Neither of these, though, shows access path details in a straightforward list. If you're like me, you are a bit frustrated with IBM for not giving us a simple, clean way to see what access paths exist over a file and what their details are. I've written a utility called Display Access Paths (DSPACP) to solve this problem. Here, I give you the details you need to use DSPACP quickly and easily.

## The Screens at a Glance

Let's look at an example. Say you have a file called MENU`MST` with `x` DDS defined logical files, `y` SQL views, and `z` SQL indexes over it. Executing the command DSPACP MENU`MST` presents the screen in Figure 1.

From here, you can easily see what key fields exist for each of the access paths. In this example, file MENU`MST` has the key fields shown on the same line as the file. Notice that file MENU`MSTL5` shows four key fields with a „+“ to the right; this means that the file has more than four keys. Press F20 to see the next four keys for all the files. If there are still more keys, press F20 until no more keys show. Pressing F19 will move back through the keys. Knowing which key fields are available is only one part of what you need to know.

You might choose a file based on its keys, but what if it also has Select/Omits or has a different record layout that doesn't contain all the fields you require? Using F11, you can see a number of other details. As Figure 2 shows, the file type tells whether you are looking at a physical file (PF), a standard DDS defined logical file (LF), a DDS defined single-format join logical file (LF-JN), an SQL view (LK-VW), or an SQL index (LF-IX). This view also shows the format name and the key type (e.g., KU Keyed Unique, AR Arrival, EV Encoded Vector Index).

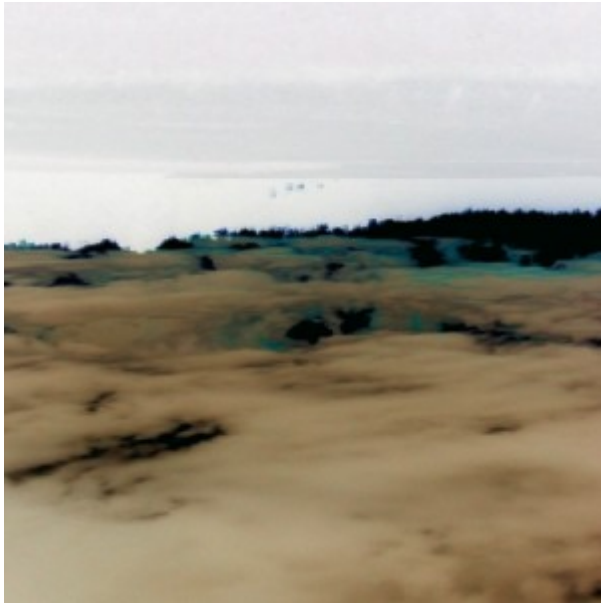
The rest of the columns show whether the file has Select/Omits and/or multiple formats, whether the format is different from the base physical, and by how many fields the access path varies from the base physical file. Using F11 again shows the description associated with the file. Most of the other function keys are pretty self-explanatory: F3 exits, F4 prompts, F5 refreshes, and F9 retrieves commands that you entered either on the command line at the bottom of the screen or via F21 (which displays a command line). The F9 key also retrieves parameter values you might have entered when list options are prompted or executed (more about prompting of options later). F10 executes DSPPFM, F12 cancels, F15 redisplay the list — first in order of key fields, then by format name, then back to the default view by physical file and access path name. F17 and F18 let you move to the beginning and end, respectively, of the list. F22 prints a list of the access paths' details.

## Der Autor

Peter Martin is an IT consultant in Sydney, Australia, with too many years in IT. He has worked for a few years in the UK but failed to develop an English accent. Currently, he works for Acumen Business Solutions, a company that provides a wide range of IT and management services and software in the government, manufacturing, and financial services sectors

## A Must-Have Utility

Now that I've written and used this utility, I find it frustrating when I have to go back to using IBM's DSPDBR and DSPFD. For me, DSPACP is simply one of those must-have utilities that all developers amass over time. It's one of my favorites, and I hope you get the same value out of it as I do.



### [Künstler Burgy Zapp](#)

The F16 key lets you search the list of files via two search screens (Figure 3 and Figure 4). Here, you can enter one or more selections to restrict the list. Most are straightforward, but the key fields selections are worth a closer look. You can enter up to three key field names; entering only the names shows only those files with all of these key fields in any position and in any order. Sometimes, though, you want to see key fields in certain positions. The number to the right of the key field name specifies which position that field must exist in. For example, where I work, most files have a company number as the first key. If, say, I am looking for transaction date, I know that the company number will be first, so I enter a 2 to indicate that I want transaction date as the second key field.

You have several options you can use against an individual access path. Option 5 executes command DSPFD, and option 8 executes DSPOBJD — straightforward enough. Option 9 shows DSPLYT, but what if DSPLYT doesn't exist on your system or in your library list? In this case, the option text shows 9=Display fields (DSPFFD), and DSPFFD executes. Later, we will see how that is controlled without changing source code. You control option 10 similarly.

Option 12 shows you a full list of the key fields for the access path in a separate screen (Figure 5), as well as other file details. If any of the files have Select/Omits, you can use option 15 to display their details (Figure 6). Option 16 displays the SQL statement used to create the access path (Figure 7). By the way, if you use option 12 on a file that has Select/Omits or SQL statements attached, you can display these with F15 and F16.

One more thing: If you don't know the physical file that an access path exists over, simply use the access path's name on the DSPACP command. The program will determine the physical file and process DSPACP appropriately. In this case, the list shows the access path at the top of the screen. Simply use rollback or F17 to go to the top of the list.

### **Load 'n' Go nur für NEWSabonnenten plus:**

November Utility2 der NEWSolutions

- Load 'n' go
- [Value Added Article: Display Access Paths Utility](#) by Peter Martin
  - Inhalte: Siehe unten

## Panel Group Options

Behind the scenes, this utility uses a CL program as the command's CPP. The CL checks for the access path's existence and sends a message to the caller if it is not found. If the access path is not the underlying physical file, it determines the physical file, which it then passes to the main processing program. An RPGLE program (DSPACPR) provides the main processing, supported by a panel group to display the results and a second RPGLE program (DSPACPR2) to process the panel group options. Two other panel groups provide help text — one for the command and the other for the display panel. Additionally, a DDS defined printer file prints the access path details. In a panel group, options can be processed in one of two ways. You can use only one method in any one list; when you define a list, you must decide which method to use. The choice is defined using the „actor“ keyword — either UIM or CALLER (or NONE if there are no options at all). Actor = UIM means that the panel group processes all the list options; CALLER means that the calling program processes them. When the User Interface Manager (UIM) is in charge, you specify commands to be executed for all of the options available. These can be any iSeries or user-written commands (e.g., DSPFD, DSPLYT) or a call to a separate program that can retrieve and process the option. You can also include substitution values in the command string. So why choose the UIM to process options instead of processing the option yourself? Here are a few reasons; the UIM

Sie müssen sich als Abonnent anmelden um den hier fehlenden Teil des Inhalts zu sehen. Bitte [Login](#) für Zugriff.

Noch nicht Abonnent? [Sonderaktion nutzen](#).

- [7 Euro/Monat NEWSabo digital - sofort zugreifen & online bezahlen.](#)
- [13,5 Euro/Monat NEWSabo plus inkl. 5x Logins & Print-Ausgaben - sofort zugreifen & per Firmen-Rechnung bezahlen.](#)