Making A Field Change? Tap Into PATHFINDER

Database fields are always being increased in size to accommodate changes in business volume such as customer numbers, invoice numbers, PO numbers, etc. New interfaces are going to be created between traditional applications and e-commerce. What files and programs will be affected? How will you begin to answer these questions?

With PATHFINDER, you have the tools needed to implement the enhancements successfully with minimal disruption. In fact, this type of project will tap many of the most powerful PATHFINDER features. Following is a description of the various features and how they will help make projects more manageable.

Before using many of the suggested options, the Object X-ref and Field X-ref databases must have been built using option"1. Build/refresh object X-ref" from the Object X-ref Menu and option "1. Built/refresh field X-ref" from the Field X-ref Menu.

General PATHFINDER Features

A quick mention of some general principals of the way PATHFINDER works is in order.

Command Entry

Each PATHFINDER option can be executed thru command entry, in addition to the menu option that we provide. While using any option in the package, the associated command name is displayed in the upper left corner of the screen. Command entry is a very comfortable environment for many programmers, making the package accessible just as easily as the operating system features. Furthermore, it is very powerful to write CL programs that issue PATHFINDER commands to perform a series of steps automatically.

Outfiling

Most PATHFINDER options can direct their results to a database file, giving you an easy way to query or massage the data depending on your specific needs. It is especially useful when making a database change because of the volume of work to be done. By gathering the results of features like Field where used or Scan source in a database file, you can easily automate steps in the process.

The output file is created as an externally described file using the name that you specify. The first way to create an outfile is to use the "F14=Submit panel" command key that is available on the entry panel of each option. The "F14=Submit panel" presents a screen where the option to outfile data can be selected and the details of the outfile can be specified. Secondly, the command entry for each option in PATHFINDER includes the OUTPUT(*OUTFILE) parameters just like the OS/400 commands do. Combined with the command entry, the

outfiling becomes a very straight-forward method of coding PATHFINDER features into CLP's that automatically perform any analysis that you need.

User-defined options

PATHFINDER implements user-defined options in the same way PDM does. For any operations that need to be performed redundantly against items in the PATHFINDER lists, consider creating a user-defined option to perform the task more easily. For example, when you have identified a file or program that will be included in the project, you can execute a user-defined option to copy the source into a testing library, thus saving you the work of typing the command on the command line for each item. The user-defined option to accomplish this would look like the following:

```
"SC" – CPYSRCF FROMFILE(&SL/&SF)  TOFILE(TESTFILES/&SF)
                FROMMBR(&SM)
```

Also note that there are two user-defined option files shipped with PATHFINDER for your use. They are HAWKEYE/USERPATH for use within PATHFINDER and HAWKEYE/USERPDM for use within PDM. Each file includes options already defined for you. Some examples that will be handy during the project will be mentioned as this article continues.

Cleanup

Anything you can do to eliminate unnecessary work will help you to stay on schedule. So before beginning the process of analyzing the applications, be sure the system is free of typical problems using a few PATHFINDER reporting tools. One company we spoke with located literally hundreds of programs that were either never used or non-functional in the first place, and obviously didn't need to be included in the project. For many companies, database changes serve as the catalyst for some long overdue cleanup.

Unreferenced objects

PATHFINDER's Object X-ref contains information about the relationships among the objects, such as what programs use what files and so on. The option "9. Unreferenced objects" on the Object X-ref Menu uses the X-ref to identify objects that exist on disk but have no references to them. These "orphaned" objects may not need to be included in the project, and indeed, may not need to be kept on the system at all.

Object last used

The operating system logs the last used date in the object description and PATHFINDER's option "6. Object last used" on the Added Time Savers Menu organizes this usage information into a very powerful report. By specifying a "range of days" you can select "dead" objects on the system that have not been used in a long time and determine if they need to be kept on the system and included in the project.

Unreferenced source

The scope of this project could force you to investigate and modify programs and files that haven't been touched in years. And through the seemingly endless cycle of hardware and software upgrades, staff turnover and contractors, you may have lost source for existing programs or accumulated "archive" copies of source from prior modifications. The result is a big mess. Using option "7. Unreferenced source" on the Added Time Savers Menu, you can identify both objects that have no source and source members that have no compiled object. This knowledge is very useful before you get too deep into the project. For example, if you are missing source for a large block of programs it will influence how you decide to address the situation. Plus, by eliminating backup copies of source members, the time spent scanning and analyzing the source can be reduced to just the necessary items.

Locating the Fields Involved

File layout

The easiest way we've found to analyze your database fields is to use option "4. File layout" on the File Analysis Menu. This option, like others in PATHFINDER, allows outfiling data in addition to its display and print capabilities. By selecting *ALL as the file and naming the database library, and outfiling the results as recommended in the previous section, you have a database of every external field known in the application. You can easily query the outfile selecting fields that follow particular naming conventions or sizes. In fact, iSeries (AS/400, Power Systems, System i) Query can outfile the results of these selections, letting you gather the suspect fields in a separate file, ready for the next step in the analysis. For many companies, this will be significantly easier than searching through the list of fields interactively.

Field X-ref by field

Option "3. Fields X-ref by field" on the Field X-ref Menu will retrieve a list of files that use a field by a particular name. Like the other options on the Field X-ref Menu, generic and wildcard entries are allowed. Use a generic selection in conjunction with the "F8=Work with" key to browse a list of field names, choosing the ones that you're interested in.

Files using reference fields

Option "7. Files using REFFLD" on the Field X-ref Menu will show any fields that are defined by a field in a field reference file. Many applications use a field reference file where a field is defined in one place and then that definition is used to define like data in an application.

Compile Lists

You will probably want to keep a record of the objects you find while using these options. We suggest using a PATHFINDER Compile List. A Compile List is a named list of objects. Use the command WRKCMPL (Work With Compile Lists) to manage the available Compile Lists as well as create new ones. And you can add items to a list using the subfile list option "13=Add to compile list" or by using the command ADDCMPLE (Add Compile List Entry). You may also use the user-defined option, "AE", that is shipped with PATHFINDER.

As you identify files that need to be included in the project, you can easily add them to the list. Later in the project, Mass Compile can recompile all the objects on the list to create a test environment as well as replace the originals at implementation time. We'll refer to Mass Compile in more detail later in this article.

Locating the Programs Involved

Now we'll use PATHFINDER to locate the programs that specifically use the fields we identified.

Field where used

The Field X-ref menu also has some "where used" options that will get a list of programs that specifically use the fields in question. You can use option "9. Field in all files where used" to search for fields based on names, and option "11. REFFLD where used" to search based on fields in the field reference file. These are very similar to the options discussed in the previous section, except these will list the programs that use these fields.

Object where used

If your application uses routines that are copied into programs using the RPG /COPY or the COBOL copybook features, use option "2. Object where used" on the Object X-ref Menu. Use object type of "*MBR" to obtain a list of programs that use a copy member.

If your application uses utility programs to perform various calculations, use option "2. Object where used" to find where these programs or commands are being referenced. One example is the TAATOOL/ADDDAT (Add Date) User Tool program.

Scan source members

Some projects involve changing non-field or non-file elements embedded in your source code.

Use option "2. Scan source member(s)" on the Added time Savers Menu. One powerful feature of Scan Source is that one job can scan multiple source physical files in multiple libraries. It also allows you to scan for up to ten different character strings in a single job. So with PATHFINDER's Scan Source option, a single job can search all your program source for both multiplication factors.

Testing the Effects of Changes

Mass Compile

With a project of this scope, you will likely want to create a full test environment that includes all the objects that are involved. Option "5. Mass compile" on the Added time Savers

Menu allows you to submit a single batch job to compile a list of both files and programs. One approach is to create all the objects into a new library, ideal for creating a test environment.

The Compile List that was discussed in the previous section can be used as input, thus telling PATHFINDER which files to compile. A strong feature of Mass Compile is that it can automatically retrieve a list of programs that use any of the files in the list. This eliminates the need to also maintain a Compile List of programs for the project. Any programs that are not included based on their relationship to one of the files, can be added to the same compile list as the files so that they will be included in the job.

Mass Compile also includes the option to have the data members and the file attributes retained on the new copies. Mass Compile will automatically create the new files and programs with the same authority as the originals but this can be overridden so the test environment is created at a programmer's level of authority.

Spacing Charts

If you must make changes to the report and display layouts, use the Spacing Charts Menu to test the effects. These options print a sample of output described in RPG o-specs, COBOL program described reports, DDS printer files and DDS display files. One noteworthy point is that it is based on the source member, so there is no need to recompile before viewing the new layout of the report or display. And, there is never a need to generate test data for these reports, they will use the standard SDA-like designations for character and numeric data including the edit word and edit code manipulations.

As mentioned earlier in this article, user-defined options are available to execute these options easily from both PDM and PATHFINDER subfile lists:

"RP" – RPG o-spec spacing chart
"CP" – COBOL defined spacing chart
"DP" – DDS printer file spacing chart
"DD" – DDS display file spacing chart

Implementing the Changes

Mass Compile

By now you have made all the source changes in your test environment and you've tested to everyone's satisfaction. You have also moved all the changed source members into production in preparation for recompiling. Just as Mass Compile allowed you to create the test environment, you can now use the Compile List to initiate a Mass Compile job replacing the original files and programs.

For this job you would tell Mass Compile to map data from the original object. This includes backing up the data before recompiling and automatically copying the data back when

done using the CPYF (Copy File) command in the operating system. Additionally, Mass Compile will preserve existing file attributes and object authorities when replacing the originals. As you can see the Mass Compile option will save plenty of time over implementing changes manually.

Wrap-Up

We have hit some of the major features of PATHFINDER that can help with ANY project including the Field X-ref, the Object X-ref, Spacing Charts and of course Mass Compile. If you would like more detailed information on any area of the package, simply contact us by email at info.hawkinfo.com or call us Monday-Thursday, 7 a.m. to 5 p.m. (MST) and Friday 7 a.m. to 3 p.m., VOICE (970) 498-9000 or FAX (970) 498-9096 at Hawkeye Technical Services. We can assist you in deciding how best to apply the various PATHFINDER features to your particular needs.