OmniFind, Part II: Integrating OmniFind Text Search Server with DB2 Web Query

Published Wednesday, 08 July 2009 01:00 by MC Press On-line [Reprinted with permission from iTechnology Manager, published by MC Press, LP; http://www.mcpressonline.com.]

Written by Gene Cobb – cobbg@us.ibm.com

Organize and secure your documents, and create a GUI to perform text searches on them.

In my previous article "OmniFind, Part I: Add Sizzle to Your SQL with OmniFind Text Search Server for DB2 for i," I introduced IBM OmniFind Text Search Server for DB2 for i , a new IBM i 6.1 product that provides the ability to perform both basic and sophisticated text searching against your DB2 for i data. In this article, I show you how to integrate this exciting new product with DB2 Web Query, the strategic IBM i query and reporting tool.

To recap Part I, the OmniFind Text Search Server supports the searching of a DB2 row based on text contained in either a database column or a document (a PDF file or XML document, for example) that is stored in a database column with a data type such as Large Object Binary (LOB). It provides two new integrated SQL functions (SCORE and CONTAINS) that you can use in your SQL statements to specify and execute text-based searches.

For example, let's say you have a DB2 for i table called INVENTORY that contains information about the all products you sell. Columns in the table include product_number, product_category, and product_name. In addition, for each product, you also have a PDF file stored in a LOB column named tech_spec_doc. Each PDF file contains all of the technical specification information for that particular product. Now, let's say that you have a requirement to find all products containing the string "headphones" in the PDF document. The "OmniFind infused" SQL statement would look something like this:

```
SELECT product_number, product_name
FROM inventory
WHERE CONTAINS(tech_spec_doc, 'headphones') = 1
```

This statement would return all PDF documents stored in column tech_spec_doc that contain the word "headphones."

While this is certainly cool stuff, you may have noticed that it lacks a graphical interface to perform these searches and actually open one of the matching documents. Well, that's where the IBM DB2 Web Query for i product comes in. DB2 Web Query pairs well with OmniFind because it provides the graphical interface needed to collect the search criteria from the end user and display results of the

OmniFind search. While there currently is not native support for OmniFind within DB2 Web Query (meaning it does not generate the OmniFind-specific syntax in the SQL statement it submits to the database engine), marrying the two technologies together can be accomplished by using stored procedures.

Behold the Stored Procedure!

A stored procedure is really just a program object that is registered to the database. It can be written in the SQL procedural language or any supported IBM i language, such as RPG, COBOL, C, or Java. You can use it to execute customized business logic and return a result set to the caller or the client application.

When it comes to DB2 Web Query data sources to base their reports on, many developers only use database objects like tables and views. But a very powerful, yet underutilized feature of DB2 Web Query is its ability to use a stored procedure as a data source, provided the stored procedure returns a result set. When a DB2 Web Query synonym is created over a stored procedure, the result set is used as the format for the synonym. This means that all fields in the result set can be used as columns in the report. In addition, the input parameters of the stored procedures can be used as input parameters for the report. Consequently, you can pass parameter values from a report to the stored procedure, let the procedure use those values to perform the business logic, and return the results to the report.

Why is this such a powerful feature? Because it gives the developer the programmatic control during the execution of the query and, consequently, the ability to do many things, one of which is to call the functions necessary to interface with the OmniFind Text Search Server. For this particular kind of implementation, the stored procedure's job is simple: accept the search criteria as an input parameter, generate the SQL statement with the correct OmniFind function syntax, execute that statement, and return the results as an SQL result set. The result set returned from the stored procedure is received by DB2 Web Query and can be used as the data source for a report.

To best illustrate how everything works together, let's continue with the example that was started in the previous article. In that example, we performed the following tasks:

1. Added a Binary Large Object (BLOB) column to a table
2. Loaded PDF documents into the BLOB column
3. Started the OmniFind Text Search Server
4. Created and updated the text index
5. Performed searches using OmniFind-specific syntax in SQL statements

To extend this example and integrate it into DB2 Web Query, we will take these additional steps:

1. Create a stored procedure
2. Create a DB2 Web Query synonym over the stored procedure
3. Create a DB2 Web Query report over the synonym
4. Implement URL drill-down

Step 1: Creating the Stored Procedure

As previously mentioned, stored procedures can be written in any supported IBM i programming language. For our example stored procedure, we will use an RPGLE program with embedded SQL. The source code for this stored procedure is shown below:

```
D MYFILE          s                   SQLTYPE(BLOB_FILE)
D SQL_FILE_OVERWRITE...
D                 C                   const(16)
d searchString    s            128a
d selectStm       s            256a
d inv_row         ds                             qualified
d  prod_num                     4a
d  prodcat                     30a
d  prodname                    60a
d  score                       10i 0
d outResults      ds                   dim(9999) qualified
d  score                       10i 0
d  prod_num                     4a
d  prodcat                     30a
d  prodname                    60a
d  docName                    200a
d counter         s             10i 0
d omf_inv         PR                   ExtPgm('OMF_INV')
d   searchString              128a
d omf_inv         PI
d   searchString              128a
 /free
   counter = 0;
   // Build the SQL statement with the OmniFind functions
   selectStm =
   'SELECT prod_num, prodcat, prodname, +
   INTEGER(SCORE(tech_spec_doc,''' +
   %trim(searchstring) + ''')*100) AS SCORE +
   FROM inventory_omf +
   WHERE CONTAINS(tech_spec_doc,''' +
   %trim(searchstring) + ''')=1 +
   ORDER BY +
   SCORE(tech_spec_doc,''' +
   %trim(searchstring) + ''') DESC';

   EXEC SQL PREPARE S1 FROM :selectStm;
   EXEC SQL DECLARE C1 CURSOR FOR S1;
   EXEC SQL OPEN C1;
  //Read each matching row and populate the result set array
   dow sqlcod = 0;
     EXEC SQL
       FETCH c1 INTO :inv_row;
     if sqlcod <> 0;
       leave;
     endif;
```

```
          counter = counter + 1;
          outResults(counter).score = inv_Row.score;
          outResults(counter).prod_num = inv_Row.prod_num;
          outResults(counter).prodcat = inv_Row.prodcat;
          outResults(counter).prodname = inv_Row.prodname;
          outResults(counter).docName =
               %trim(inv_Row.Prod_num) + '.pdf';

          // Detach the document to a directory in the IFS
          MYFILE_NAME = '/DB2WebQuery/OmniFind/' +
               outResults(counter).docName;
          MYFILE_NL = %len(%trim(MYFILE_NAME));
          MYFILE_FO = SQL_FILE_OVERWRITE;
          EXEC SQL
            SELECT tech_spec_doc INTO :MYFILE FROM inventory_omf
                   WHERE prod_num =  :inv_Row.prod_num;
        enddo;

        // Return the result set
        EXEC SQL
         SET RESULT SETS ARRAY :outResults FOR :counter ROWS;
        EXEC SQL CLOSE C1;
        return;
      /end-free
```

This program does several things for OmniFind integration:

- Accepts one input parameter: the string to perform a text-based search over
- Builds the SQL statement with OmniFind syntax for the search string
- Prepares the SQL statement, declares a cursor for that statement, and opens the cursor to execute the statement
- Reads each matching row and adds it to the result set array
- Extracts (detaches) the document in the BLOB column for each matching row to the IFS directory named /db2webquery/omnifind (in case the user would like to open that document later)
- Returns the array to the client application (DB2 Web Query in this case) as a result set once each matching row has been read and the result array has been built

To create this program, issue the following command:

```
CRTSQLRPGI OBJ(MYLIBRARY/OMF_INV_SP)COMMIT(*NONE)
```

Once the program has been created, the next step is to register it to the database as a stored procedure. You do this by using either an SQL interface (such as the STRSQL command) or the Run SQL Scripts window in System i Navigator and specifying the following SQL statement:

```
CREATE PROCEDURE MYLIBRARY.OMF_INVENTORY_SP (
      IN SEARCHSTRING CHAR(128) )
      LANGUAGE RPGLE
```

```
NOT DETERMINISTIC
READS SQL DATA
CALLED ON NULL INPUT
EXTERNAL NAME 'MYLIBRARY/OMF_INV_SP'
PARAMETER STYLE GENERAL ;
```

To test the stored procedure, simply call it from the Run SQL Scripts window interface in System i Navigator and specify your search string as the input parameter:

```
CALL OMF_INVENTORY('Stereo')
```

Step 2: Creating a New Synonym Based on a Stored Procedure

Now that you have a stored procedure that is successfully returning a result set based on OmniFind search syntax, you are ready use it in a DB2 Web Query report. The first thing you must do for any DB2 Web Query report is create a synonym over the data source. As mentioned, a stored procedure is a valid data source. To define the stored procedure synonym, take the following steps:

1. Open a Web browser and log into DB2 Web Query.
2. Expand the domain and folder you want to place your report in.
3. Right-click on the folder and select Metadata (Figure 1).

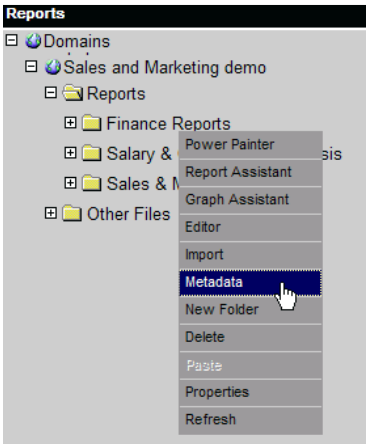

Figure 1: Access your metadata.

Another Web browser is launched, and the DB2 Web Query metadata wizard is presented.

4. Under Adapters, expand DB2 cli, right-click *LOCAL, and select Create Synonym (Figure 2).

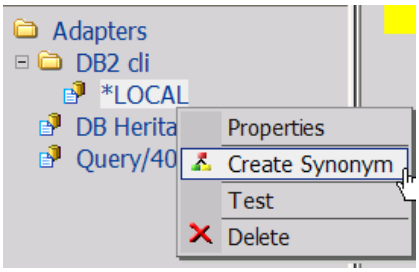

Figure 2: Start creating your synonym.

5. On the Restrict Object Type To drop-down list, select Stored Procedures. The example stored procedure was created in library MYLIBRARY, so specify that library and click Next (Figure 3).

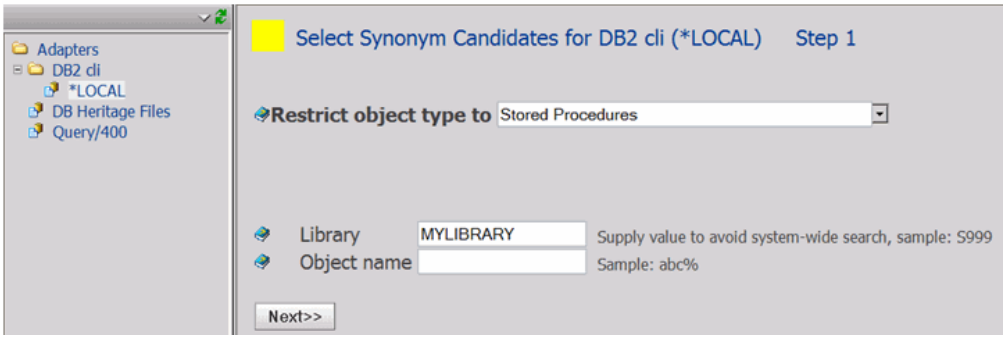

Figure 3: Choose Stored Procedures as object type.

6. A list of stored procedures in that library is presented. Select the OMF_INVENTORY stored procedure and click Next (Figure 4).

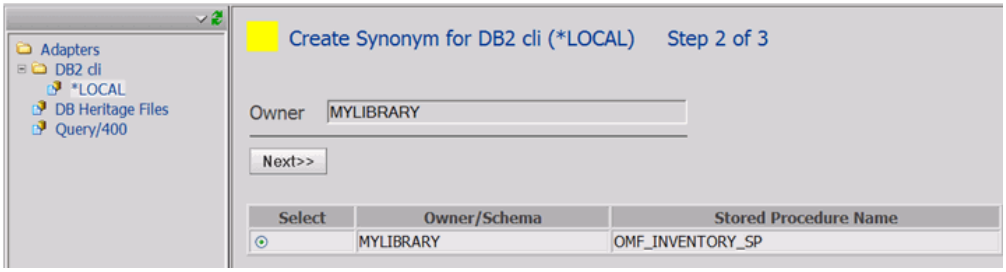

Figure 4: Select OMF_INVENTORY.

7. Do the following:
   - Select/check the SEARCHSTRING parameter.
   - Specify 'TEST' as the value for the input parameter.
     Note: Any string value will work, so just type 'TEST'. When the synonym is created, the stored procedure is actually called by DB2 Web Query so that it can receive the result set. This is because it needs to store the format of the result set in the synonym. Therefore, you need to pass it a value for the input parameter at this step.
   - Click Create Synonym (Figure 5).

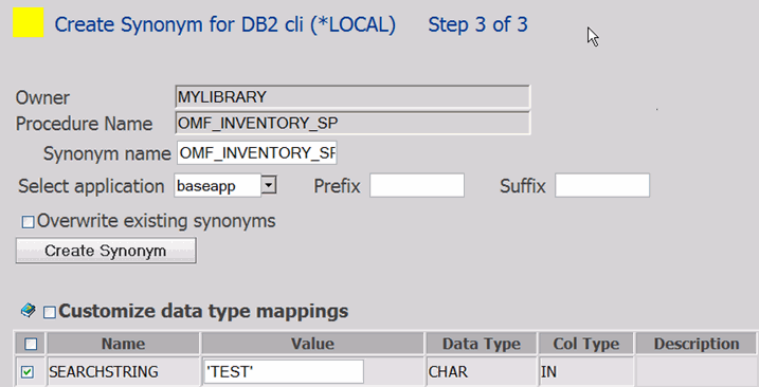

Figure 5: Create the synonym.

8. A confirmation window is displayed. Click Close.

You have successfully created a synonym for the stored procedure. On to the next step!

Step 3: Creating a Report Based on the Stored Procedure Synonym

With the stored procedure synonym in place, you are now ready to create a new report.

1. Back in your first browser session, right-click on your folder and select Report Assistant (Figure 6).

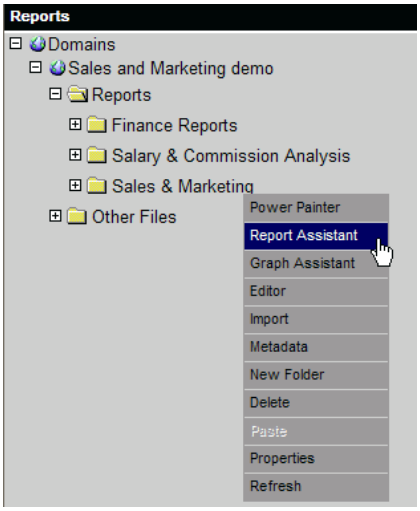

Figure 6: Access the Report Assistant.

2. From the list of synonyms displayed, select OMF_INVENTORY_SP, the stored procedure synonym you just created (Figure 7).

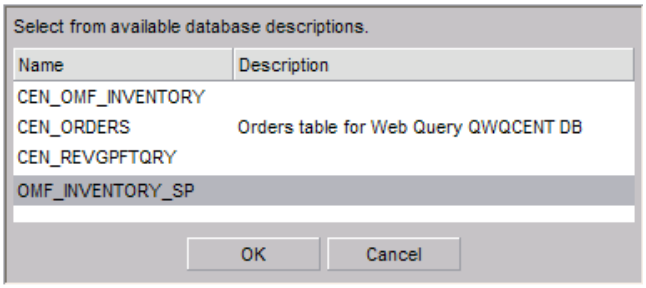

Figure 7: Select the OMF_INVENTORY_SP stored procedure synonym.

The Report Assistant tool is presented. You see a list of all of the columns of the stored procedure result set as well the stored procedure's input parameter (Figure 8).

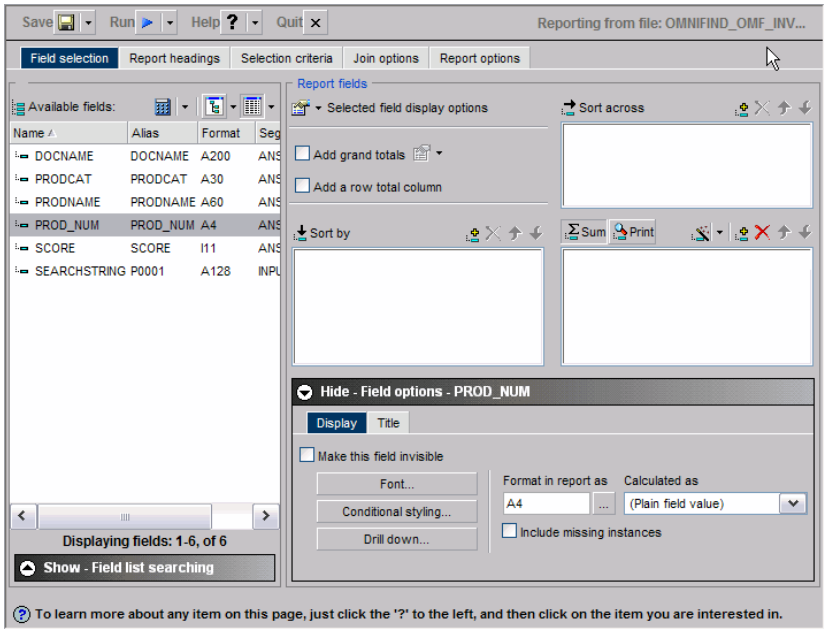

Figure 8: Your Report Assistant tool looks like this.

3. From the list of available fields, drag SCORE into the Sort By pane.
4. While SCORE is highlighted in the Sort By pane, click the icon at the bottom left to show field options (Figure 9).

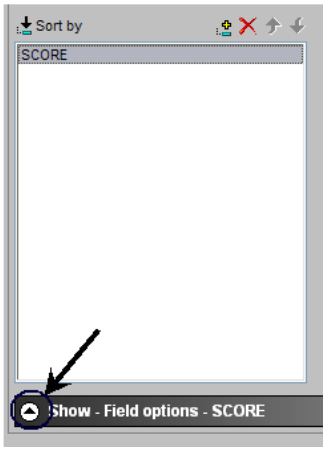

Figure 9: Show field options.

5. For the SCORE column, click the Sorting tab and select Descending order. This will display the most relevant matches first (Figure 10).

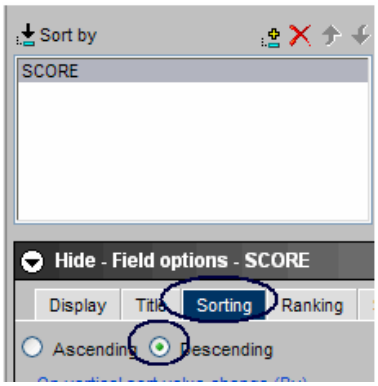

Figure 10: Sort by descending order.

6. Do the following:
   - Drag the PRODCAT field into the Sort By pane.
   - Drag the PRODNAME field into the Sort By pane.
   - Drag the PROD_NUM field into the Sum pane.

When you're finished, the Field Selection tab of your report should look something like Figure 11:

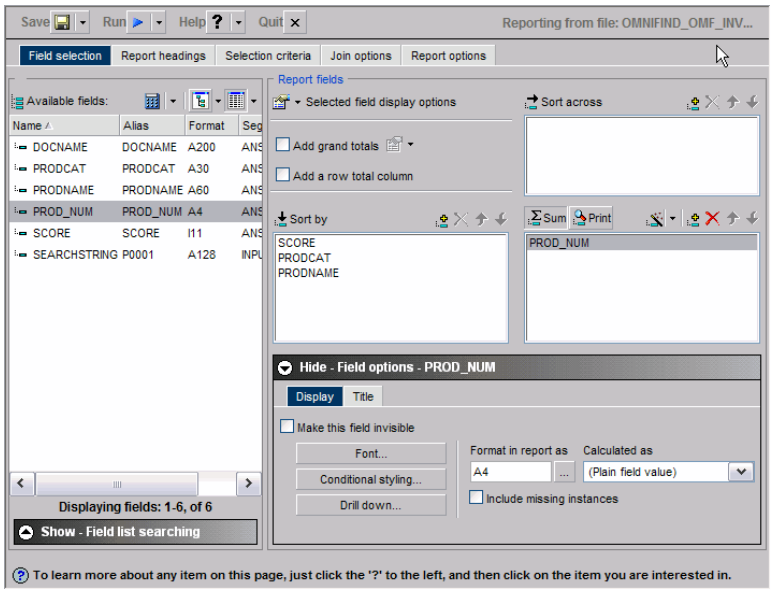

Figure 11: Now, your Field Selection tab shows fields.

Next, you need to define one input parameter: the search string that the user will specify and that the report will pass to the stored procedure.

7. Click on the Selection Criteria tab and do the following (Figure 12):
   - From the list of available fields, drag SEARCHSTRING into the Screening conditions pane.
   - Accept the default condition Equal To.
   - Click on the <Selected values> link.
   - Select Parameter and specify INSTRING as the parameter name.
   - Click OK twice to save this parameter definition.

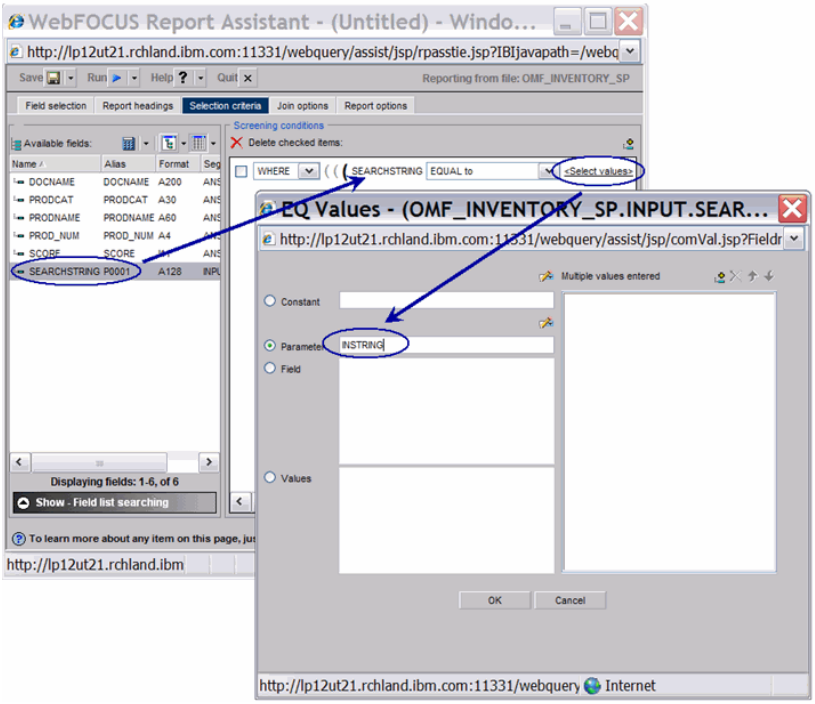

Figure 12: Select your criteria.

8. Save the report as "OmniFind Inventory Search" and Quit to close the Report Assistant dialog window.
9. Your new report should now be displayed in the folder. Go ahead and run it!
10. Specify "stereo" for the INSTRING parameter and press the Run button (Figure 13).

SCORE	PRODCAT	PRODNAME	PROD_NUM
17	Amplifiers/PreAmps/Tuners	PA4000 Stereo & Surround Power Amplifier	1002
14	Amplifiers/PreAmps/Tuners	Modular Components Series Preamp 5.1	1003
12	CD Players and Recorders	CD Recorder with 50GB Hard Disc Drive	1022
9	CD Players and Recorders	Digital CD Turntable	1024
		Multichannel Super Audio CD Player	1025
		400 Disc Super Audio CD Changer	1023
8	Audio Systems	Micro HiFi Stereo System	1011

Figure 13: Run your stereo report!

The OmniFind Text Search Server returned seven documents (stored in the INVENTORY table) that matched the search string "stereo." These matching rows are sorted in descending order by their "score" (a relevance value that is based on how well the document matches the search argument).

Step 4: Implementing a URL Drill-Down

You may be perfectly satisfied with this report and be tempted to declare it finished. But one of your users will ultimately ask if the matching documents can be opened directly from the DB2 Web Query report. You may already know that DB2 Web Query has the ability to link to another Web application through the browser. This is known as "URL drill-down," and you can use this technique to similarly drill down into a document that is located on the IFS. If you have a network drive mapped to the IFS of the IBM i or have a file share set up for the IFS directory that contains the documents, you can open files on the IFS using a Web browser.

For example, let's say that you have a file share named MyDocs that points to the IFS path /home. If a PDF file named resume.pdf is stored in that directory, you can open this file from the browser by specifying the following URL:

```
\\ip_address\mydocs\resume.pdf
```

This is the naming format we will use to drill into the PDF documents from the report.

Setting Up a File Share

For this example, we are going to set up a file share using the System i Navigator.

1. Launch System i Navigator and open a connection to your IBM i system. Expand File Systems > File Shares and select New > File Share (Figure 14).

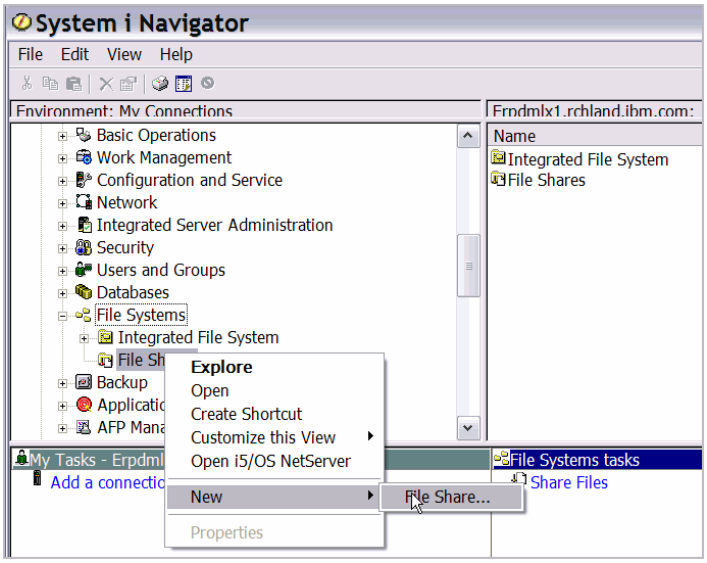

Figure 14: Start setting up a file share.

2. The Create i5/OS NetServer File Share dialog window is displayed. From this window, specify the following (Figure 15):
   - Share name: OmniFindDocs
   - Access: Read only

- Path name: /DB2WebQuery/OmniFind

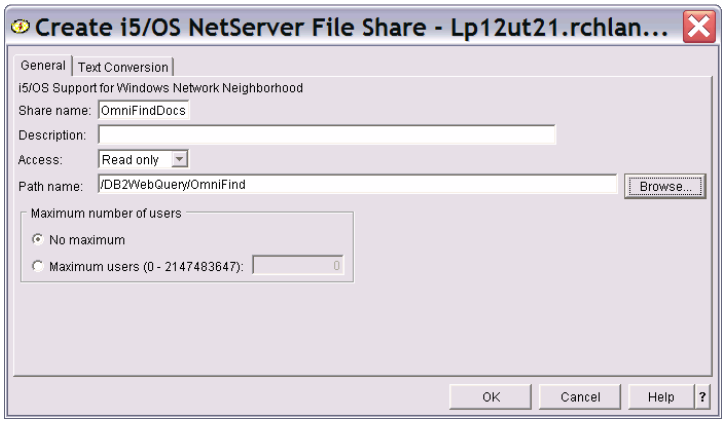

Figure 15: Define your file share.

3. Click OK

Creating a Defined Field for URL Drill-Down

The final step is to create a defined field and use it for the basis of a URL drill-down. This will allow the user to click on a column in the report and open the underlying document from within a DB2 Web Query browser session.

1. Return to your DB2 Web Query browser session and open the OmniFind Inventory Search report. This will open the report in a new Report Assistant session and allow you to make some modifications.
2. Click on the down arrow next to the icon that looks like a calculator and select New Defined Field.
3. Name the new defined field DRILLDOWN, specify A300 for the Format, and enter the following expression (substituting your system name or IP address): '\\your_ip_address\omnifinddocs\' || DOCNAME. See Figure 16.

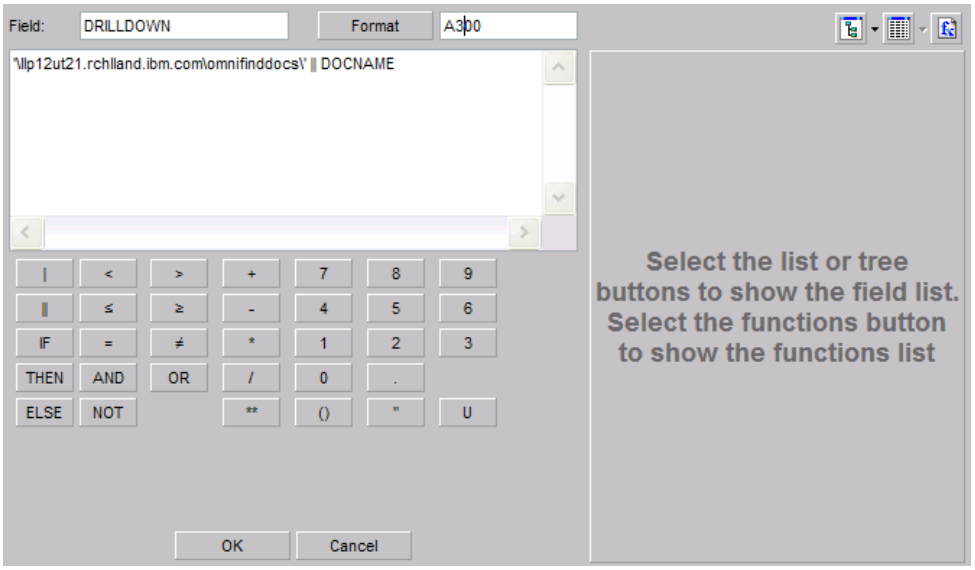

Figure 16: Create a defined field.

Note: DOCNAME is the column that contains the name of the PDF document. This is obviously needed if we want to open the document.

4. Click OK.

While we do not want to display the DRILLDOWN column in the report, it must be included so that it can be referenced in a drill-down definition. We can simply add it to the report but make it invisible.

5. Drag the new DRILLDOWN field into the Sum pane. Hide it by selecting Make This Field Invisible (Figure 17).

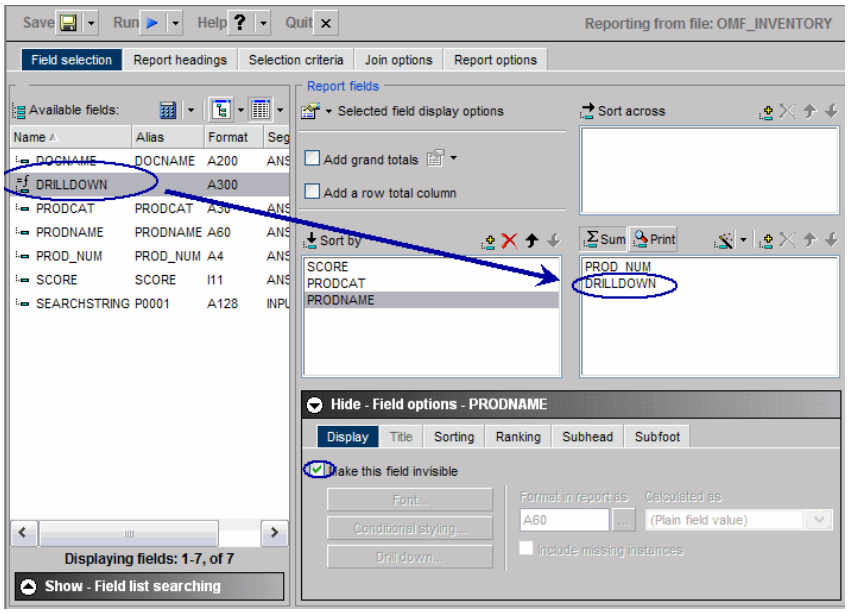

Figure 17: Drag DRILLDOWN into the Sum pane.

For this report, we would like the product number column to be the "hotspot." That is, when the user hovers over that column, a hand appears, indicating that the user can click on that column to drill down and open the underlying document. This is done by defining a drill-down for the PROD_NUM column.

6. Under the Sum pane, select PROD_NUM, and click on the Drill Down button (Figure 18).

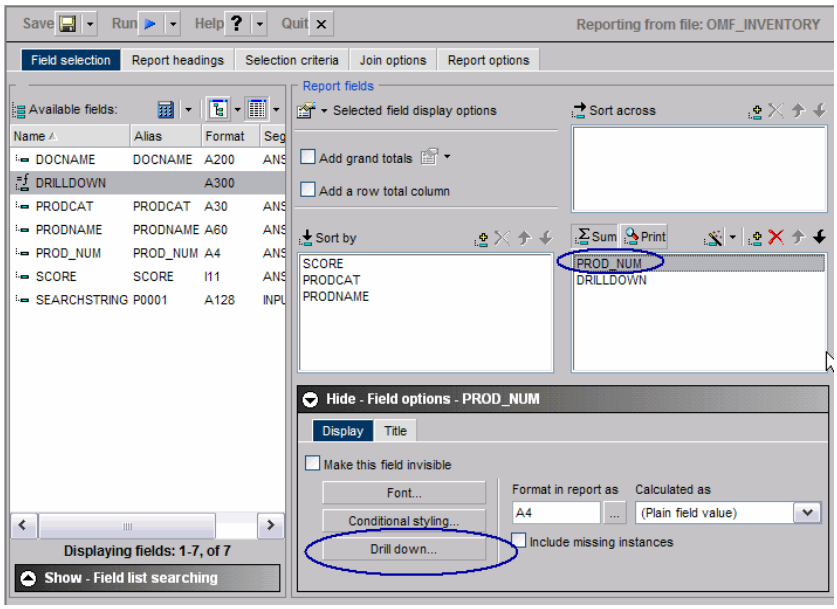

Figure 18: Drill down into PROD_NUM.

7. Do the following (Figure 19):
   - Select Execute URL.
   - For the URL, specify (DRILLDOWN). Note: The specification of the defined field (DRILLDOWN) is case-sensitive. So ensure that it exactly matches the defined field name.
   - Click OK.

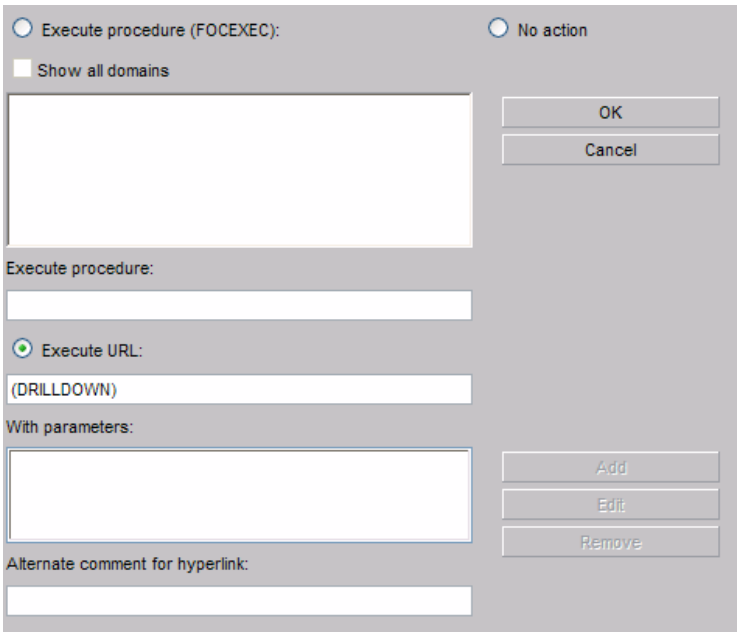

Figure 19: Execute the (DRILLDOWN) URL.

8. Click the Run button to test the report. Another browser window will open.
9. Specify "stereo" for the search string and click Run (Figure 20).

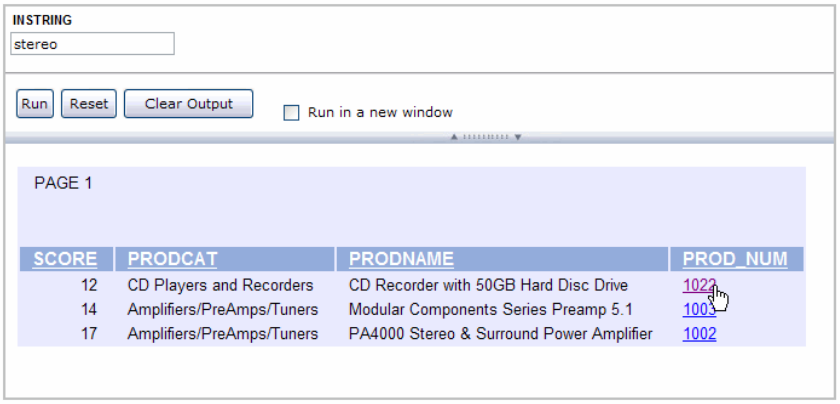

Figure 20: Run the stereo report.

Notice that the PROD_NUM columns are now hotspots.

10. Click on one of the product numbers to drill down into the PDF document (Figure 21).

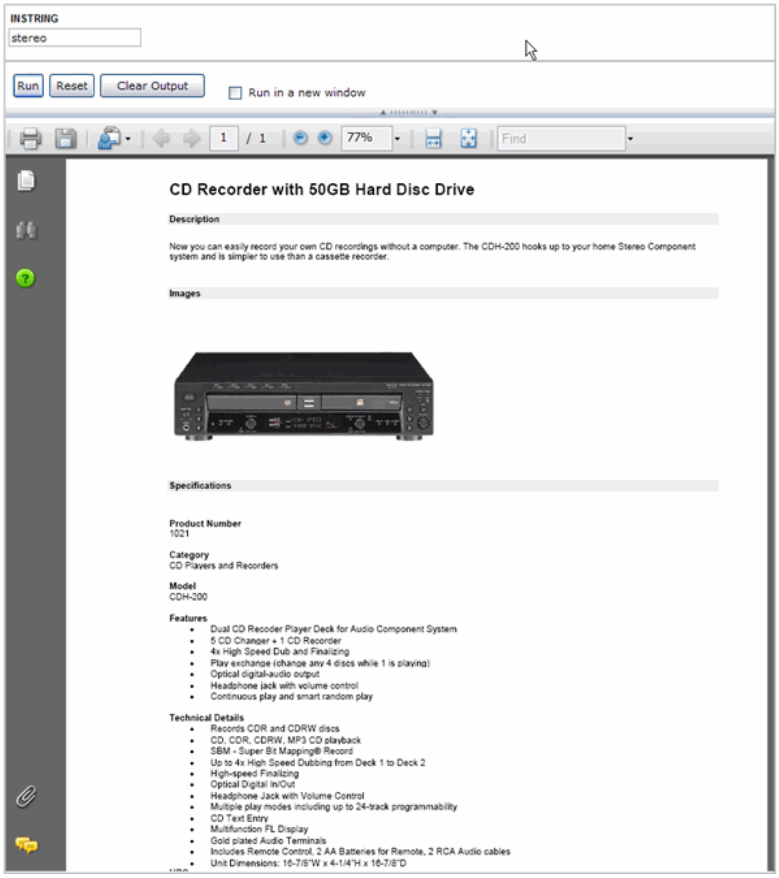

Figure 21: Drill down into the PDF doc.

Return to the Report Assistant session and save your report as OmniFind Inventory Search.

Security Considerations

If you have concerns about exposing sensitive data that is contained in your documents, consider this: DB2 Web Query users must log into the system to identify themselves. Consequently, when they run a report that uses a stored procedure synonym, the underlying program runs with that user's authority. If you have object-level security or row-level security (using SQL views), you will not be exposing sensitive data. Always remember that no DB2 for i interface (including DB2 Web Query) can circumvent object-level security.

If you are concerned about the risk of exposing sensitive data in the documents detached by the stored procedure, you could set up a more secure environment than the one provided in the example above. For example, you could create and secure an IFS directory for each individual DB2 Web Query user and have the stored procedure detach the documents into the current user's directory. For example, for user profile JSMITH, the stored procedure would detach all documents into the path /db2webquery/omnifind/jsmith/, and access to the path would be granted to just that profile. Other user profiles (unless they had *ALLOBJ authority) would not be able to access that directory to get at those detached documents.

You may have other ideas about how to design a secure environment. That's the beauty of utilizing a stored procedure data source: because you have programmatic control, you can implement whatever design you want!

The Search Is Over...

Hopefully, this two-part article has given you some ideas on how to use the OmniFind and DB2 Web Query products in ways that will dazzle your end users. You may not need all of the features that were covered, but if your IT shop needs one or more of the following...

- A way to centralize, organize, access, and secure your documents
- A graphical interface that is easy to create and use and can perform both simple and sophisticated text searches over your documents
- An interface to open the documents

...you can either develop your own application or use these techniques to deliver a solution that is easy to implement and provides the features you are searching for!