Get a Handle on Metadata

Published online September 2009 as part of the IBM Systems Magazine

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

In the September IBM Systems Magazine, Power Systems—IBM i edition cover story, "Untangling Web Query", I explained why DB2 Web Query uses metadata and how it can actually reduce the complexity of your query and reporting environment and make life easier for your report developers. In that article, I told you about the benefits of using metadata. In this article, I want to share more details on how to take advantage of those benefits with more hands-on details.

What Is Metadata?

To refresh your memory, metadata is simply data about data. DB2 Web Query metadata is a materialized repository that contains information about your database files. Before you can create a report or graph in DB2 Web Query you must first create metadata (also referred to as synonyms) over the data sources. You can create a synonym over such database objects as tables or physical files, SQL views, DDS logical files, stored procedures and materialized query tables.

You can generate metadata in three ways:

- Create your own. Metadata creation wizards are available from both the Web browser via the Web Query launch page and in the DB2 Web Query Developer Workbench tool. To create metadata, you must be a Web Query administrator (member of group profile MRADMIN) or a developer in the domain. This is the most common way metadata is created.
- Use third-party tools. Three good examples of vendors who've enabled their technology for DB2 Web Query are Databorough's x-WebQuery, Information Builders' iWay Data Migrator and Coglin Mills' RODIN DB2 Web Query Edition.
- Find an ERP or services provider. ISVs have the capability to include DB2 Web Query content as part of their solutions package. If you purchase such a distribution, the metadata will be included in the package. Similarly, a services provider may have the expertise and tools to build the metadata for you.

Synonym Editor

In the previous article, I shared that the DB2 Web Query Developer Workbench tool, a Windows PC-based tool, offers several powerful components, one of which is the Synonym Editor. The Synonym Editor contains graphical interfaces that let you:

- Define database joins
- Build virtual columns to centralize business logic

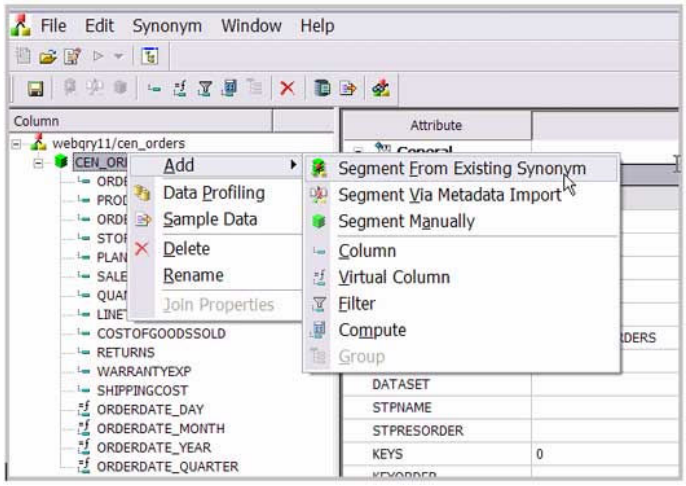
- Standardize column formats
- Convert and standardize date fields
- Create filters
- Create business views
- Define online analytical processing (OLAP) dimensions

You read the benefits of these. Now let me walk you through how to use them. For more information on how synonyms affect the IFS, read the sidebar, “Two Stream Files.”

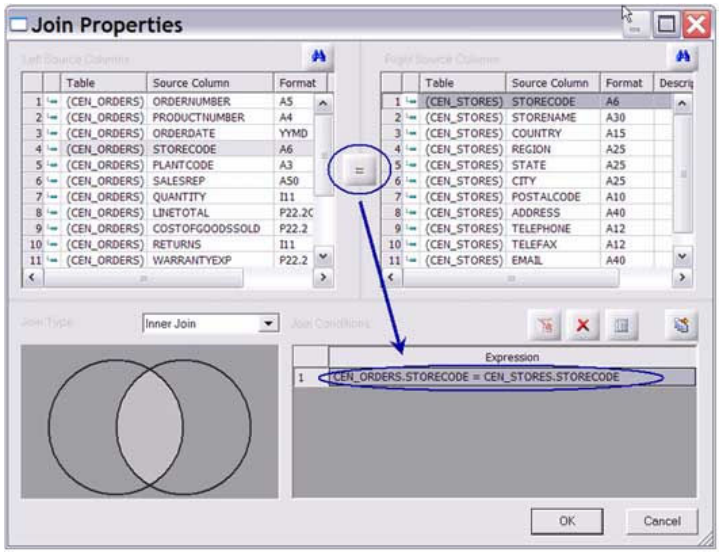
Define Database Joins

If you’ve developed reports with Query/400, you’re aware that if the query requires more than one file, you must define your joins in each query definition. Metadata lets you define your joins in one place and base all of your reports on that single, centralized definition.

Using the Synonym Editor in Developer Workbench, you can add new join segments from existing synonyms or files that simply need to be brought into the cluster (and don’t require standalone synonyms). Figure 1 shows how to add a new join segment.

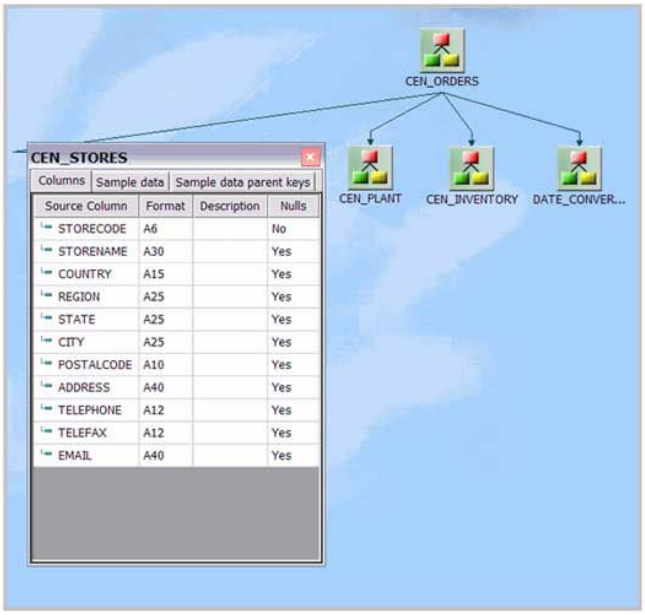


Once you’ve added a join segment to a synonym, you must define the join fields by right-clicking on the new segment and selecting “Join Properties” which results in a dialog box where you can select the join fields from each segment as shown in Figure 2.



When you're finished adding all your join segments, the Synonym Editor will group the columns under their respective segments and provide an interface from which the segments can be expanded and collapsed as shown in Figure 3.

The Synonym Editor also provides a graphical representation of the model in the Modeling View tab (see Figure 4). You can double-click on any of the segments to show the column information and sample data for that particular file.



After saving the synonym, use the DB2 Web Query development tools such as Report Assistant and select the synonym as the data source. When Report Assistant dialog is presented, all of the columns in each of the join segments can be used in the report. Because the join segments have been defined in the synonym, you won't need to define them again in the report.

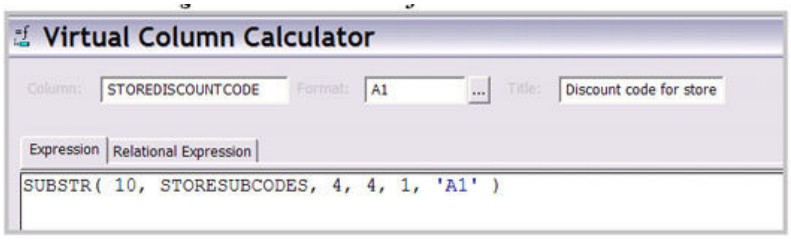
Build Virtual Columns to Centralize Business Logic

Centralizing business logic that's specific to your company into a DB2 Web Query metadata layer helps eliminate errors and developers' need to know every formula for every output.

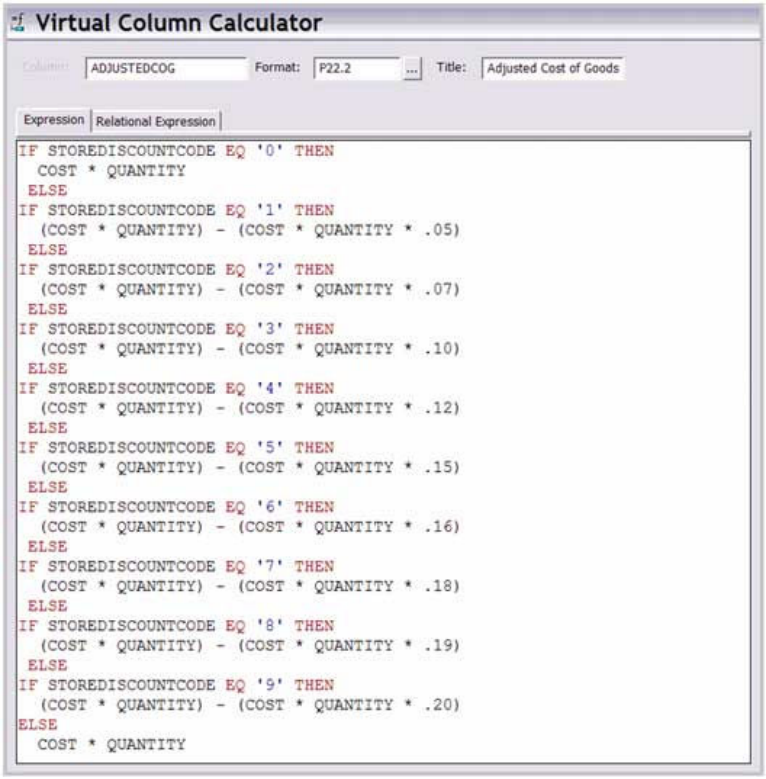
Instead, calculated and conditional fields can be abstracted in DB2 Web Query by creating virtual columns in the synonym. For example, let's suppose there are two joined tables, ORDERS and STORES. The STORES table contains a field named STORESUBCODES, which is a 10-character string with each character representing something different for that particular store. The fourth character of this field is used to calculate order discounts. The valid values are 0-9 and each value represents a specific discount rate. A value of zero means the store receives no discount, and value of 9 means the store is granted a 20 percent discount. Any value between those two equates to a specific discount somewhere in between that range.

If you wanted to develop a series report that calculates measure such as gross profit for each store, you'd have to know and understand this business rule and apply it to every report. It'd be much easier to perform this in the metadata so it only must be defined once.

To do this, you'd create two virtual columns in the metadata. The first one, STOREDISCOUNTCODE, extracts the fourth digit from the STORESUBCODES field (see Figure 5).



The second virtual column would contain the expression to apply the appropriate discount (based on the value of STOREDISCOUNTCODE) and calculate the adjusted cost of goods sold (see Figure 6).



The adjusted cost of goods sold column can now be easily added to any report to accurately calculate values such as gross profit without needing to redefine this specific business logic across dozens of various profit reports.

Standardize Column Format

If your reports include numeric currency columns such as revenue and gross profit, you'll likely require some standard formatting of these values. You can define this in the metadata so you don't have to repeat the formatting definitions across hundreds of reports. The Synonym Editor provides an interface to easily select these formatting specifications. I've shown an example in Figure 7.

Attribute	
General	
FIELDNAME	LINETOTAL
ALIAS	LINETOTAL
MISSING	✔
TITLE	Revenue
ACTUAL	P11
Type	Decimal Packed
Length	11
Decimal	0
USAGE	P22.2CM
Type	Decimal Packed
Length	22
Decimal	2
Options	
Negative	Default - Minus sign at the left: -6148
Comma	C - Insert: 41,376
L - Leading zeroes	
S - Print blank for zero	
% - Percent sign	
Currency Symbol	M - Floating $ sign

Convert and Standardize Date Fields

I've worked with many IBM i shops that store their date values in a data type that isn't a date or timestamp, often because the database was created prior to the availability of these data types. Consequently, it's common to find databases with packed decimal and character fields that actually store date values. An example of these legacy date fields is a field defined as packed decimal (8,0), which contains the value 06252009 to represent the date June 25, 2009 (see Figure 8).

	ORDDAT
1	1262007
2	3192007
3	2152007
4	3032007
5	2282007
6	3242007
7	2022007
8	1022007
9	2142007
10	3212007

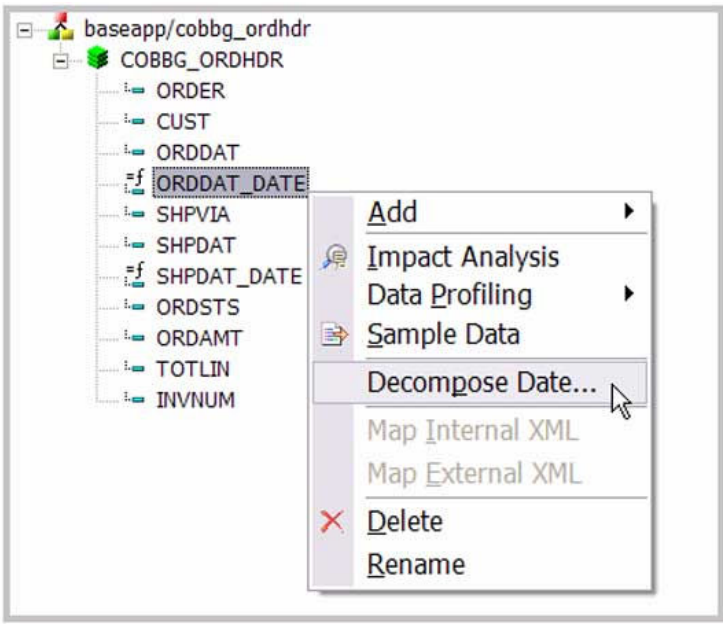
Because they aren't true date fields, they can be problematic for DB2 Web Query: it simply doesn't know these values actually represent dates. As you can see in Figure 9, to the tool these fields are just regular packed decimal fields.

Attribute	
General	
FIELDNAME	ORDDAT
ALIAS	ORDDAT
MISSING	
TITLE	
ACTUAL	P5
Type	Decimal Packed
Length	5
Decimal	0
USAGE	P9
Type	Decimal Packed
Length	9
Decimal	0

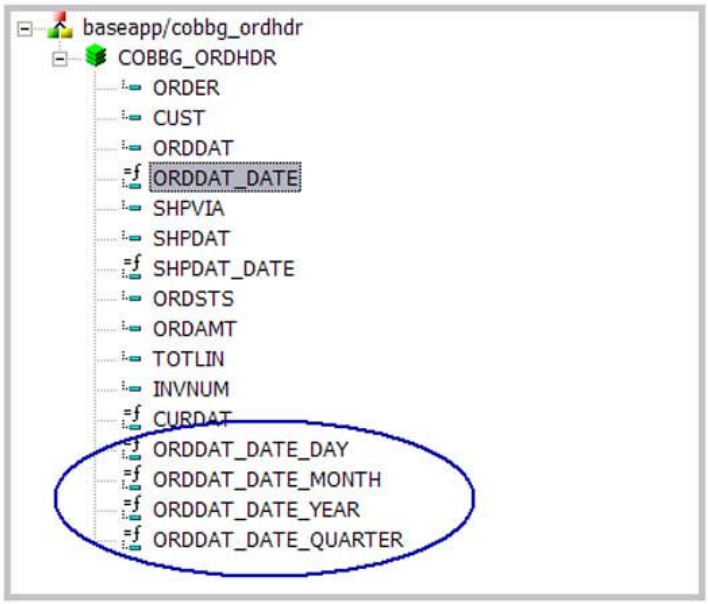
While this may be fine for storing the date values, it's probably not the format you want dates displayed in your reports. DB2 Web Query can convert the legacy date field using a new virtual column that's a true date field. See an example in Figure 10.

Attribute	
General	
DEFINE	ORDDAT_DATE
EXPRESSION	DATECVT(ORDDAT, 'P8MDYY', 'MDYY')
TITLE	Order date
FORMAT	MDYY
Type	Date
Date Components	Three-part: YYMD, MDYY, DMYY
Date Order	MDYY - 08/21/2004
Year	YY - 4 digit year - 2004
Month	M - 2 digit month
Day of the Week	
Separator	/ - 2004/08/21
DEFCENT	0
YRTHRESH	0

Once you have a true date field defined, you can decompose the date field. This means breaking down the date field into four new and separate virtual columns: year, quarter, month and day. From the right-click menu, select the date column and then Decompose Date (see Figure 11).

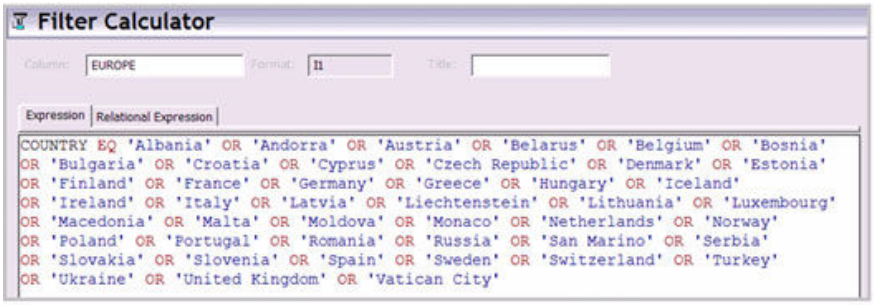


As you can see in Figure 12, the synonym now contains the four new virtual columns that can be used in your reports.



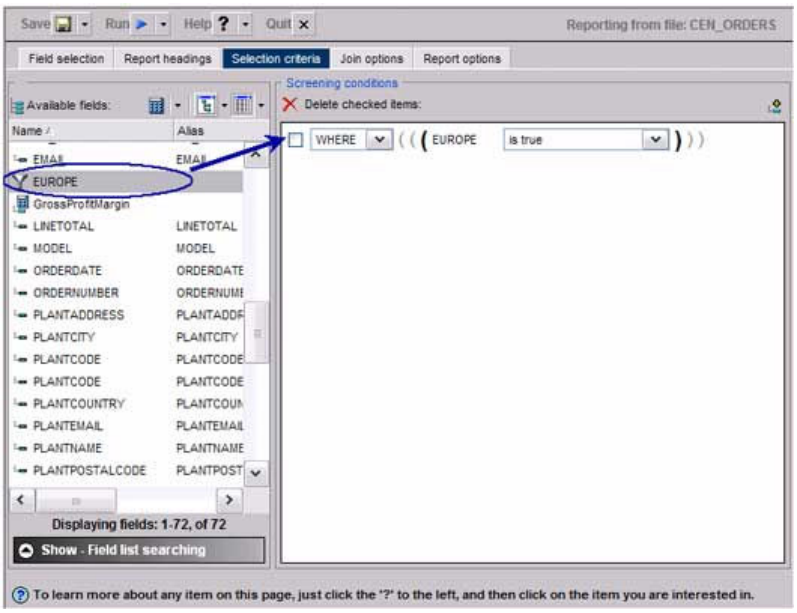
Create Filters

I also provided the example in my first article about creating reports for all of the countries in Europe and told you that DB2 Web Query filters can eliminate complexity by allowing the metadata administrator to define all the European countries in the synonym. Then you can create a filter that contains each of the countries in Europe. It would look something like Figure 13.



Filters are selection criteria (translated to SQL WHERE statements) that are stored in the synonym and provide the report developer with quick and easy access to the predefined selection criteria limited the data included in a report.

Then, for any report based on that synonym, the new filter appears in the list of available field, and you can simply drag it into the screening conditions pane under the Selection criteria tab as shown in Figure 14.



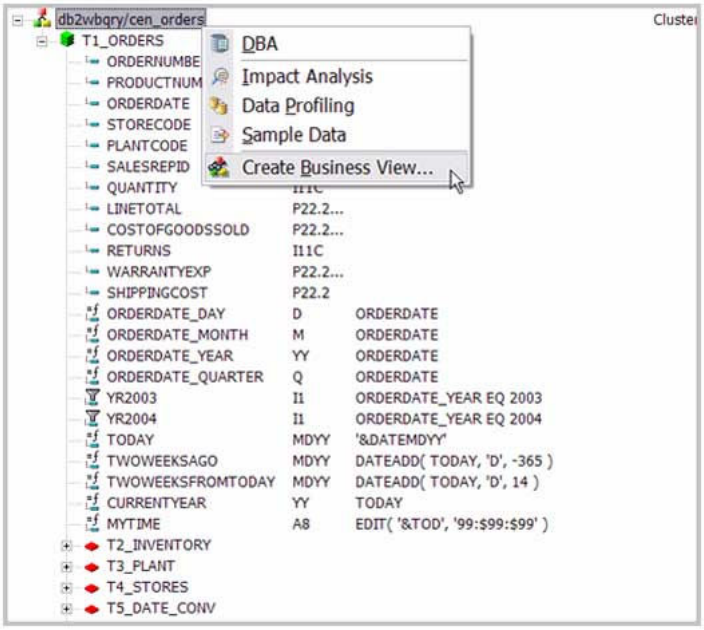
Run the report to show only orders for the European customers. Figure 15 shows an example. Your report developers will be amazed with that simplicity.

Country	Product Type	Revenue
France	Audio	$16,817,832.00
	Camcorders	$19,806,592.00
	Cameras	$8,138,513.00
	Office	$1,338,048.00
	Video	$19,381,067.00
Germany	Audio	$20,850,439.00
	Camcorders	$24,332,946.00
	Cameras	$9,601,694.00
	Office	$1,841,043.00
	Video	$29,582,533.00
Spain	Audio	$16,458,158.00
	Camcorders	$21,066,201.00
	Cameras	$6,149,115.00
	Office	$1,670,051.00
	Video	$24,134,124.00

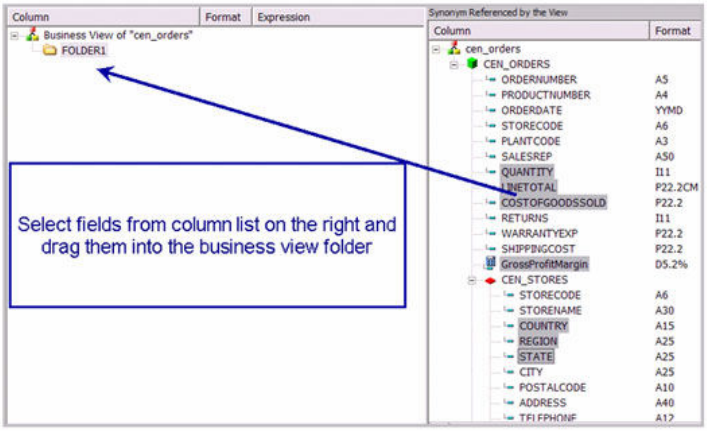
Create Business Views

Business views let you select and organize relevant columns, which is a separate synonym that can be selected as the report's data source.

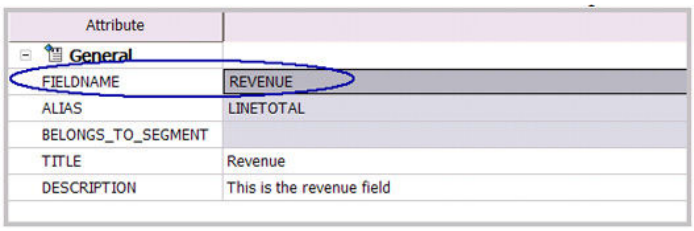
To create a business view, open the synonym in the Synonym Editor and select Create Business View from the right-click menu as shown in Figure 16.



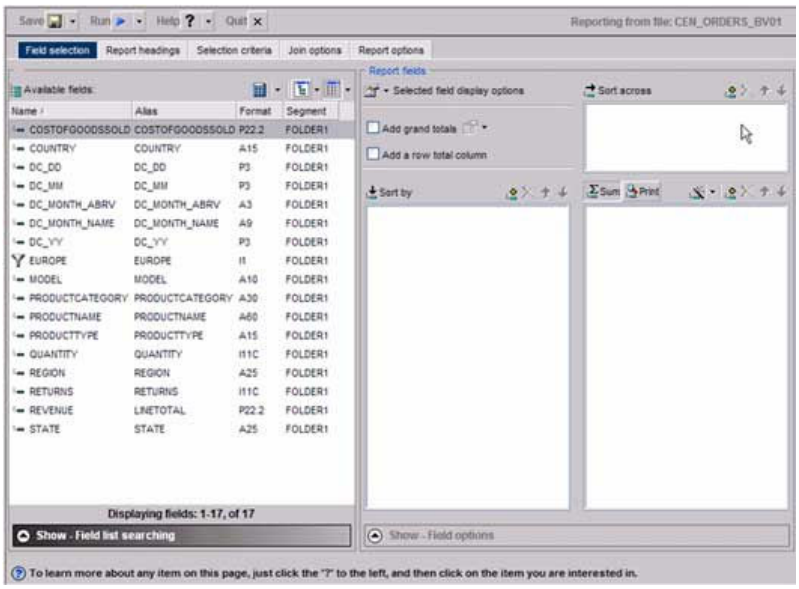
In the resulting dialog box, select the columns from the source synonym and drag them over to the business view folder (see Figure 17).



You can even rename the fields in the business view and change the column title and description. Figure 18 shows the column LINETOTAL was renamed REVENUE and a title and description were specified.

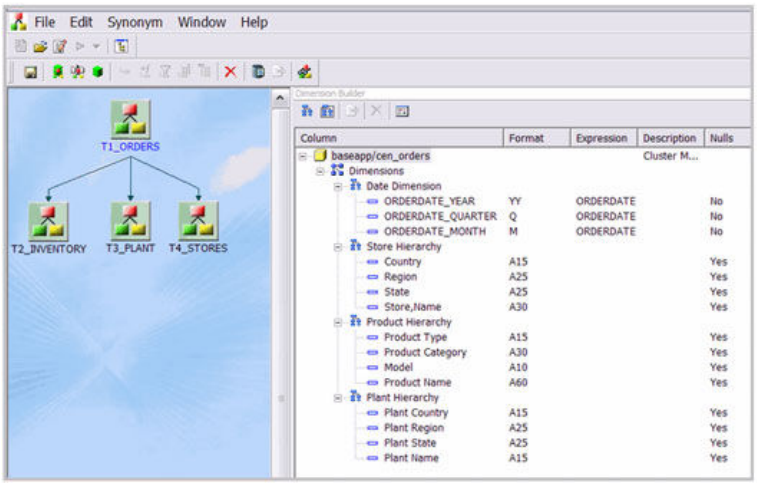


When a report is based on a Business View synonym, the report developer no longer has a list of many available fields to choose from. Instead, smaller set is presented like Figure 19.

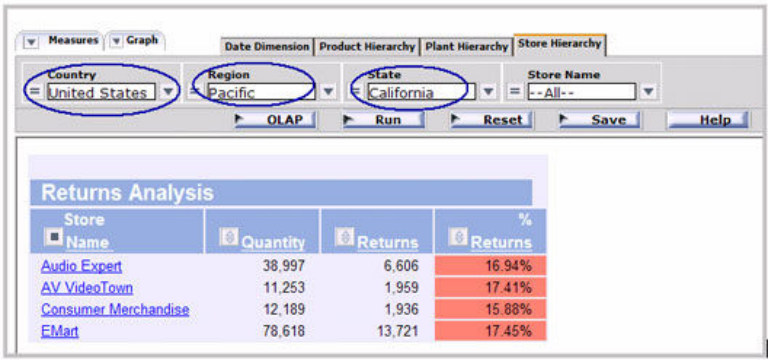


Define OLAP Dimensions

OLAP dimensions can be defined in the metadata so you don't have to create a copy of the data that's built in a multi-dimensional or cubed format. From the Modeling View tab of the Synonym Editor (Figure 20), you can define the dimension and drag columns from the synonym into each of those dimensions.

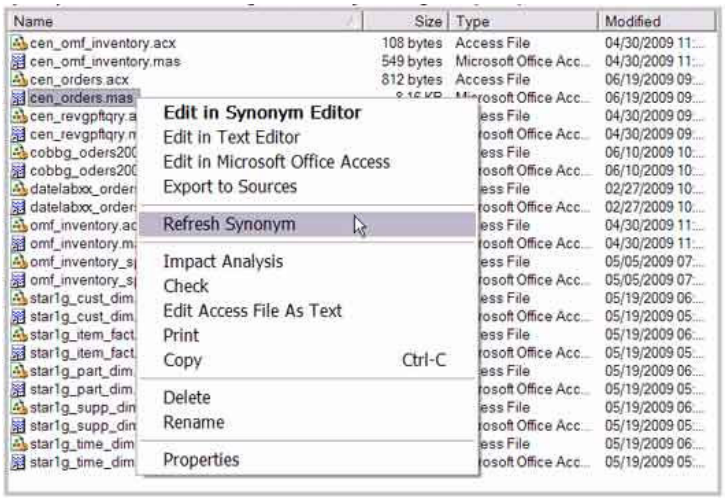


Once your OLAP dimensions are set up, you can transform very simple reports into high-powered and feature-rich OLAP reports. Figure 21 shows a report where the user has drilled down through the Store dimension to display all of the stores in California.



Refreshing Metadata

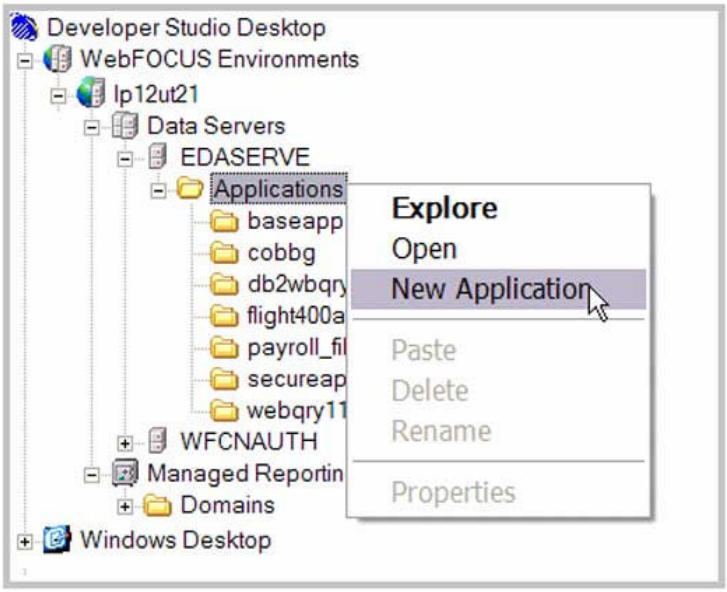
If you change the underlying data source by adding or removing columns, you may need to refresh the synonym. The DB2 Web Query Developer Workbench offers an option from the right-click menu to refresh the synonym (Figure 22). This parses the synonym and compares its contents with the structure of the underlying data source. New columns are added to the synonym and deleted columns are removed. Any custom changes (e.g., joins, virtual columns, filters, etc.) made to the synonym are preserved.



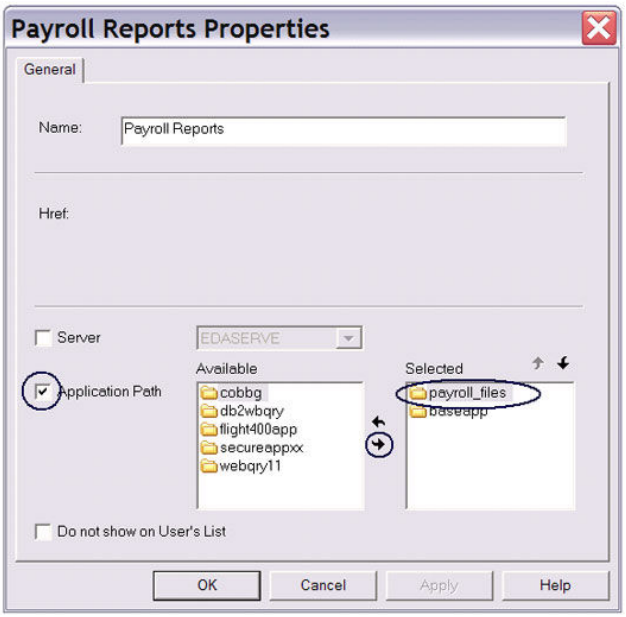
Metadata Segmentation and Security

A recent enhancement to DB2 Web Query is the capability to segment and secure your metadata. Previously, all synonyms were created in the common application folder BASEAPP, which meant they were displayable and selectable from all report development interfaces (across all domains). You couldn't filter the list of synonyms for specific users or groups of users. Now you can create custom application folders and associate them with specific domains. You can now set up the application folder (and the synonyms within it) as an extension of the domain. Because domains are secured using IBM i group profiles, you can segment and secure your synonyms.

From the Developer Workbench, select Applications under the Data Servers folder and, from the right-click menu, choose New Application as shown in Figure 23.



Once your new application is created, you simply create new synonyms in that application folder. To associate the application folder with a specific domain, select Domain Properties from the right-click menu. In the resulting dialog box, check the Application Path setting, select the desired application folder, and click the right-arrow icon between Available and Selected list as shown in Figure 24.



Learn More

Learning more about metadata can help you improve efficiency and report accuracy without requiring your developers to have a deep understanding of the data model. For more information on DB2 Web Query, I suggest the IBM Redbooks publication “Getting Started with DB2 Web Query for System i” (SG247241, www.redbooks.ibm.com/abstracts/sg247214.html).