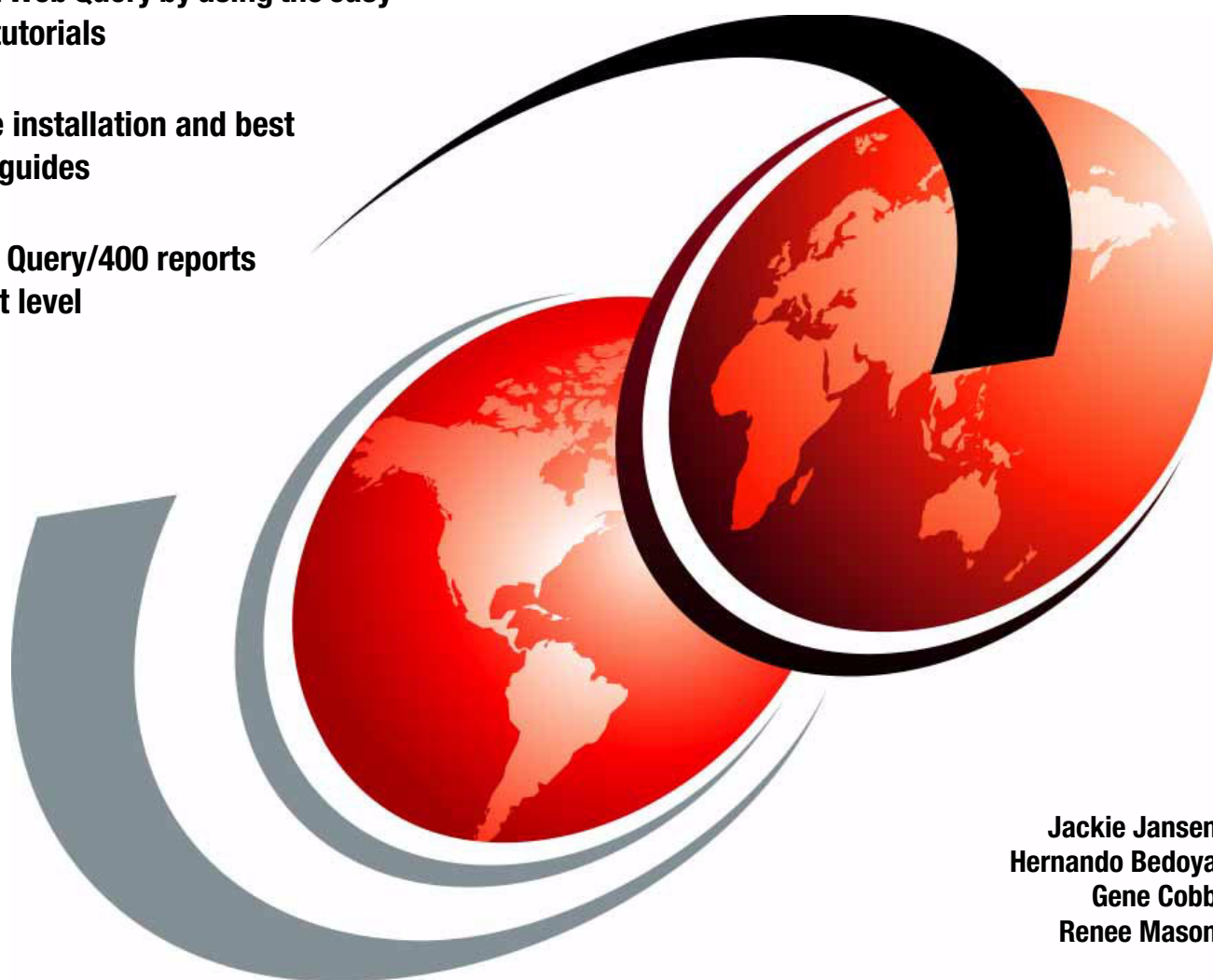


Getting Started with DB2 Web Query for System i

Learn DB2 Web Query by using the easy
to follow tutorials

Follow the installation and best
practices guides

Take your Query/400 reports
to the next level



Jackie Jansen
Hernando Bedoya
Gene Cobb
Renee Mason

Redbooks



International Technical Support Organization

Getting Started with DB2 Web Query for System i

November 2007

Note: Before using this information and the product it supports, read the information in “Notices” on page ix.

First Edition (November 2007)

This edition applies to Version 5, Release 4 of IBM i5/OS (product number 5722-SS1).

© Copyright International Business Machines Corporation 2007. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	ix
Trademarks	x
Preface	xi
The team that wrote this IBM Redbook	xii
Become a published author	xiii
Comments welcome	xiv
Part 1. Background, installation, and setup	1
Chapter 1. Product architecture and overview	3
1.1 Query/400: A product history	4
1.2 Introduction to DB2 Web Query for System i	5
1.3 Synonyms	5
1.4 DB2 Web Query features	6
1.4.1 Base features	6
1.4.2 Optional features available from IBM	7
1.4.3 Optional features available from Information Builders	8
1.5 Architecture	10
1.5.1 Web browser clients	11
1.5.2 Web server	11
1.5.3 Application server	11
1.5.4 Reporting Server	12
1.5.5 Data adapters	13
1.6 DB2 Web Query Developer Workbench	14
1.7 DB2 Web Query product positioning	14
Chapter 2. Installation and server operations	17
2.1 Installation and setup	18
2.1.1 Installing DB2 Web Query	18
2.1.2 Authorizing and verifying users	19
2.1.3 License keys	20
2.2 Requirements	21
2.2.1 PC requirements	21
2.2.2 System i requirements	22
2.2.3 Developer Workbench requirements	22
2.3 Reporting Server commands	23
2.4 DB2 Web Query server jobs	23
Chapter 3. DB2 Web Query fundamentals	25
3.1 Registering named users	26
3.2 Logging in	26
3.3 Administrator tasks	28
3.3.1 DB2 Web Query domains	28
3.3.2 Creating subfolders	30
3.3.3 Assigning authority to users	31
3.4 Developer tasks	32
3.4.1 Creating metadata	34
3.4.2 Setting up a cross-system join	45

3.5 Basic user	53
Part 2. Tutorials for DB2 Web Query	55
Chapter 4. Getting started with the tutorials	57
Chapter 5. Report Assistant	63
5.1 Tutorial overview	64
5.2 Report creation	64
5.2.1 Creating a summary report (RA1_Revenue)	65
5.2.2 Defining a report layout	65
5.2.3 Sum and detail reports	68
5.2.4 Date formatting (RA2_XTab)	75
5.2.5 Selection criteria	76
5.2.6 Cross-tab report (RA2_XTab)	79
5.2.7 Creating a sample detail report (RA3_GP)	80
5.2.8 Conditional styling (traffic lighting)	84
5.2.9 Using variables in report headings and footings	86
5.2.10 Generating subtotals	88
5.2.11 Sorting by an aggregate field	90
5.2.12 Adding ranking columns (RA4_Advanced)	91
5.3 Output options	92
5.3.1 On-demand paging	93
5.3.2 PDF output	94
5.3.3 Excel output	95
5.3.4 DB2 database file output	96
5.4 Additional report types	98
5.4.1 Parameterized reports	98
5.4.2 Drill-down reports: Parent report (RA5_Child)	102
5.4.3 Report properties	105
5.5 Joining tables (reference only)	106
5.6 Report Assistant summary	108
Chapter 6. Graph Assistant	109
6.1 Tutorial overview: Bar chart	110
6.2 Creating a simple bar chart with drill-down capabilities (GA1_Bar)	112
6.2.1 Adding multiple bars to one chart	114
6.2.2 Adding a drill down to a chart	116
6.3 Creating a dual axis bar and line chart (GA2_Parm)	118
6.3.1 Adding a second Y axis	121
6.4 Creating a revenue trend over time chart (GA3_Line)	125
6.5 Creating a line graph	126
6.5.1 Adding a date range filter to a line graph	133
6.5.2 Adding a user-specified date range parameter	138

6.6 Working with pie charts (GA4_Pie)	140
6.7 Creating a product type pie chart for each year	140
6.8 Conditional styling or traffic lighting a graph	146
Chapter 7. Power Painter	153
7.1 Tutorial overview (PP1_PDF)	154
7.2 Launching Power Painter and tailoring the main window	154
7.3 Building a graph with Power Painter	157
7.4 Building a report with Power Painter	165
Part 3. Tutorials for DB2 Web Query optional features	173
Chapter 8. Active Reports	175
8.1 Overview and highlights of Active Reports	176
8.2 Creating a report with Active Report	177
8.3 Using an Active Report	181
8.4 Summary	192
Chapter 9. Developer Workbench	193
9.1 Configuring Developer Workbench	194
9.2 Setting Developer Workbench default options	198
Chapter 10. Online analytical processing	201
10.1 OLAP terminology	202
10.2 Defining OLAP metadata using Developer Workbench	203
10.3 Enabling an OLAP report (OL1_Revenue)	210
10.4 Working with an OLAP-enabled report	212
10.4.1 Slicing, dicing, and drilling down for more details	212
10.4.2 Manipulating the data using the OLAP Control Panel	224
10.4.3 Grouping data into percentiles	231
Chapter 11. HTML Layout Painter	233
11.1 Compound parameterized report (HTML1_KPI)	234
11.1.1 Adding parameters to existing reports	234
11.1.2 Designing the layout for the HTML report	239
11.2 Highly parameterized report (HTML2_Parm)	248
11.2.1 Creating the report in HTML Layout Painter	251
Chapter 12. Additional features of Developer Workbench	265
12.1 Synonym Editor	266
12.1.1 Date Decomposition	269
12.1.2 Additional date manipulation	271
12.1.3 Creating filters	273
12.1.4 Joining tables	275
12.1.5 Business View	283
12.1.6 Impact Analysis	285
12.1.7 Data profiling	288

12.2 SQL Wizard	291
12.3 Summary	298
Part 4. Miscellaneous or additional topics	299
Chapter 13. Using DB2 Web Query to run existing Query/400 reports	301
13.1 Query/400: A reliable reporting tool	302
13.2 DB2 Web Query versus Query/400: Function similarities	302
13.3 DB2 Web Query versus Query/400: Function differences	303
13.4 Creating metadata from Query/400 objects	304
13.5 Using DB2 Web Query to edit a QRYDFN	307
Chapter 14. Performance considerations	309
14.1 Performance basics	310
14.2 DB2 CLI adapter performance	310
14.2.1 Report request process flow	310
14.2.2 Adapter processing and optimization	312
14.2.3 Remote database access considerations (including cross-system joining)	322
14.3 Query/400 adapter performance	324
14.4 DB2 for i5/OS optimization	325
14.4.1 Database design	325
14.4.2 Indexes	326
14.5 Performance case study	327
14.5.1 Identifying a long running report	328
14.5.2 Performing analysis and looking for optimization disablers	328
14.5.3 Determining report tuning options	330
14.5.4 Creating an SQL view and synonym	330
14.5.5 Creating a new report based on the SQL view	332
14.5.6 Performing additional database analysis and tuning	333
14.6 Performance benchmark	335
14.6.1 Objectives	335
14.6.2 Scenarios	335
14.6.3 Database and system configuration	336
14.6.4 Metrics	336
14.6.5 Conclusions	339

Chapter 15. Frequently Asked Questions	341
15.1 General announcement or product information questions	342
15.2 Installation and setup questions	344
15.3 Security-related questions	345
15.4 Named-user question	346
15.5 Metadata questions	347
15.6 Query/400 migration questions	350
15.7 Report development questions	351
15.8 Excel spreadsheet integration questions	356
15.9 Add-on product questions	358
15.10 Save and restore questions	360
15.11 Education questions	361
Part 5. Appendixes	363
Appendix A. Metadata in the integrated file system	365
Appendix B. Date and time functionality	367
Date and time system variables	368
Date format	368
Date format display options	368
Controlling the date separator	370
Using date fields	370
Date fields in arithmetic expressions	370
Converting date fields	371
Date functions	371
DATEADD: Adding or subtracting a date unit to or from a date	371
DATEDIFF: Calculating the difference between two dates	372
DATEMOV: Moving the date to a significant point	372
Appendix C. Processing differences between the define and compute fields	375
Appendix D. Setting up DB2 Web Query to run in an IASP environment	377
On the first system	378
On the second system	381
Appendix E. Additional material	385
Locating the Web material	385
Using the Web material	385
System requirements for downloading the Web material	385
How to use the Web material	386
Related publications	387
IBM Redbooks	387
Online resources	387
How to get IBM Redbooks	387
Help from IBM	388
Index	389

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AS/400®
Domino®
DB2®
i5/OS®
IBM®

iSeries®
Lotus®
Rational®
Redbooks®
Redbooks (logo) ®

System i™
System i5™
WebSphere®

The following terms are trademarks of other companies:

Oracle, JD Edwards, PeopleSoft, Siebel, and TopLink are registered trademarks of Oracle Corporation and/or its affiliates.

Adobe Reader, PostScript, Adobe, Acrobat, and Portable Document Format (PDF) are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

Java, JavaScript, JavaServer, JavaServer Pages, JRE, JVM, J2EE, J2SE, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

ActiveX, Excel, Expression, Internet Explorer, Microsoft, PowerPoint, SQL Server, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Pentium, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Preface

The DB2® Web Query for System i™ product is a Web-based query and report writing product that offers enhanced capabilities over the IBM® Query for iSeries® (also commonly known as Query/400) product. IBM DB2 Web Query for System i includes Query for iSeries technology to assist customers in their transition to DB2 Web Query. It offers a more modernized, Java™-based solution for a more robust, extensible, and productive reporting solution. DB2 Web Query provides Report Assistant, Graph Assistant, and Power Painter. Using these tools, customers can modernize existing Query for iSeries reports while providing a foundation for building more complex business intelligence applications, such as online analytical processing (OLAP), data mining, dash boarding, or data warehouse implementations.

DB2 Web Query provides capabilities to query or build reports against data that is stored in DB2 call-level interface (CLI) databases through the latest browser-based user interface technologies. It enables you to build new reports with ease through the Power Painter or Report Assistant and Graph Assistant components. It simplifies the management of reports by allowing you to leverage parameterized reporting. It delivers data to users in many different formats, including spreadsheets, PDF, HTML or through browser support of the Java-based thin client interface.

DB2 Web Query offers features to import Query/400 definitions and enhance their look and functionality with Power Painter or Graph Assistant. It enables you to add OLAP-like slicing and dicing to the reports or to view reports in disconnected mode for users on the go. It provides an interface to all data in IBM i5/OS® through either DB2 or Open Query File native adapters that automatically identify the files to be accessed and import the metadata into DB2 Web Query.

This IBM Redbook publication gives a broad understanding of the new DB2 Web Query product. It entails a group of self-explanatory tutorials to help you get up to speed quickly. Overall this book is designed for IT users. You can extract and use Part 2, “Tutorials for DB2 Web Query” on page 55, and Part 3, “Tutorials for DB2 Web Query optional features” on page 173, as stand-alone tutorials for anyone who is developing their own queries. Much of Chapter 8, “Active Reports” on page 175, is appropriate for those who work disconnected from the System i environment. In addition, you can use much of Chapter 10, “Online analytical processing” on page 201, if you will simply be running OLAP-enabled reports.

The team that wrote this IBM Redbook

This book was produced by a team of specialists from around the world working at the International Technical Support Organization (ITSO), Rochester Center.



Hernando Bedoya is an IT Specialist at the IBM ITSO, in Rochester, Minnesota. He writes extensively and teaches IBM classes worldwide in all areas of DB2 for i5/OS. Before joining the ITSO more than seven years ago, he worked for IBM Colombia as an IBM AS/400® IT Specialist doing presales support for the Andean countries. He has 24 years of experience in the computing field and has taught database classes in Colombian universities. He holds a master degree in computer science from EAFIT, Colombia. His areas of expertise are database technology, application development, and data warehousing.



Jackie Jansen is a Senior Certified Consulting IT Specialist with 30 years at IBM. Jackie currently works in the IBM Americas Advanced Technical Support Solutions Centre. Jackie is the author of a variety of published columns including the recent "Business Intelligence with Jackie" and "Jackie's Forum". Jackie is a frequent speaker at System i Technical Conferences and User Group meetings. You can contact Jackie by sending e-mail to jjansen@ca.ibm.com.



Gene Cobb is a DB2 for i5/OS Technology Specialist in System i Solutions Enablement. His current responsibilities include providing consulting services to System i developers, with a special emphasis in application and database modernization. He has worked on IBM Midrange systems since 1988, with 10 years in the IBM Client Technology Center (CTC), in IBM Rochester. While in the CTC, he assisted customers with application design and development using RPG, DB2 for i5/OS, CallPath/400, and Lotus® Domino®.



Renee Mason has been a Support Engineer for Database on System i for three years. She started working with U.S. clients only, but has branched out to supporting clients world-wide. She holds a Bachelor in Management Information Systems degree and a minor in Computer Science from Winona State University in Winona, MN. Her area of expertise within DB2 is journal management. She also trains other System i Support Engineers from around the world who come to Rochester to build their DB2 technical skills as part of the International Internship Program.

Thanks to the following people for their contributions to this project:

Thomas Gray
James Hansen
Joanna Pohl-Miszczyk
Jenifer Servais
ITSO, Rochester Center

Robert Andrews
Michael Cain
Fernando Echeveste
Tyler Even
Doug Mack
Jarek Miszczyk
Scott Moore
Brian Muras
Chuck Pence
Sandy Ryan
Rick Saltness
Dan Toft
Eric Will
IBM Rochester

Wilfried Blankertz
IBM Germany

Julie Maw
IBM UK

Simona Pacchiarini
Francesca Zoccheddu
IBM Italy

Larry Eiss
Robert Fricano
Gary Goldberg
Pat Powers
Terry Schwarz
Information Builders Inc.

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You will have the opportunity to team with IBM technical professionals, Business Partners, and Clients.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you will develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our Redbooks® to be as helpful as possible. Send us your comments about this book or other Redbooks in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

ibm.com/redbooks

- Send your comments in an e-mail to:

redbooks@us.ibm.com

- Mail your comments to:

IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400



Part 1

Background, installation, and setup

In this part, we introduce DB2 Web Query and provide guidance on how to install DB2 Web Query. We also introduce and provide background on IBM DB2 Web Query for System i. In addition, we provide installation and setup instructions for the product.

This part includes the following chapters:

- ▶ Chapter 1, “Product architecture and overview” on page 3
- ▶ Chapter 2, “Installation and server operations” on page 17
- ▶ Chapter 3, “DB2 Web Query fundamentals” on page 25



Product architecture and overview

Today's information overload requires companies to analyze data quickly and easily in formats that suit the needs of different types of users. Executives want quick access to track key performance indicators; financial analysts want to analyze corporate performance to evaluate profitability or meet compliance guidelines; business managers need up-to-the-minute sales and inventory information to ensure customer demand and sales quotas can be met. The information needs to be delivered quickly and easily, with robust security in place, and distributed to a multitude of different client technologies, including browsers and spreadsheets, or integrated into line-of-business (LOB) applications.

In this chapter, we introduce IBM DB2 Web Query for System i (DB2 Web Query). This comprehensive Java-based business intelligence tool provides graphical, Web-based query and report writing functions.

We include the following key topics in this chapter:

- ▶ A history of the Query/400 product including its current limitations
- ▶ An overview of DB2 Web Query including how it addresses the limitations of Query/400
- ▶ DB2 Web Query architecture
- ▶ DB2 Web Query product positioning

1.1 Query/400: A product history

When the AS/400 was first released in June of 1988, one of the optional licensed programs was the Query/400 product, 5722-QU1. This product was used by midrange programmers, administrators, and users to retrieve, format, and analyze information from AS/400 database files to produce reports. The tool was widely popular within the AS/400 community due to its compact design, intuitive interface, and robust functionality. Users created custom reports by specifying the files, record selection criteria, fields to sort and group by, and column totals from a limited number of simple panels. The reports could be written directly to a panel, to a spooled file for printing, or to another database file for further processing. Then the definition could be saved to a permanent AS/400 system object (object type *QRYDFN), so that the report could be shared with other users and run again later. With a proper knowledge of the database, reports could be generated in minutes.

The licensed program has since been renamed to IBM Query for System i and is still available, supported, and widely used on the System i platform today. However useful Query/400 has been to users in the past, it has several major shortcomings:

- ▶ Lack of a graphical interface

Recently, significant emphasis was given to “modernize” System i applications. This refers to providing a more modern (graphical) interface for tools, applications, and any generated output. Many reporting tools available today have a Web browser or rich client-based interface. They also provide the ability to generate the reports in a variety of graphical formats such as PDF, HTML, XML, and Microsoft® Excel® spreadsheets. Query/400 does not support any of these interfaces.

- ▶ Lack of support for non-DB2 for i5/OS databases

Query/400 can only access information that is stored in DB2 for i5/OS. If you require your Query/400 reports to include information from other databases, such as Oracle® or SQL Server™, then you must import these tables into DB2 for i5/OS. This method is simply not practical if you have large amounts of data that is dynamic in nature.

- ▶ Limited ability to view the data in a different way or further dissect the data and drill down

Once run, the Query/400 output is static in nature and cannot be easily manipulated for further dissection and analysis. If you want to look at the data in a different way, such as to resort the list by a different column, you must create and run additional reports. Further, the ability to select a particular element in the report and drill down for more information and analysis simply does not exist. Again you would have to run another report to obtain this information. All this results in analysis that is disruptive, lacks flow and fluidity, and is time consuming in nature. It also places a strain on an IT department that is continually writing various versions of reports to satisfy user requirements.

- ▶ Not a pure SQL interface

Query/400 is a proprietary interface and does not use the industry standard Structured Query Language (SQL) when accessing database objects. This limitation places restrictions on the functionality of the reports that can be created. New SQL functions and features that are introduced with each release cannot be directly or easily incorporated into Query/400 reports.

- ▶ Not strategic

IBM has clearly stated that SQL is the strategic database interface for both creating and accessing database objects. Since Query/400 is not an SQL interface, IBM will not invest in this technology in the future, and therefore the product will not address any of the previously listed deficiencies.

- Does not use the new SQL Query Engine (SQE)

SQE was introduced in V5R2 of i5/OS and offers algorithms and features that give it distinct advantages over its predecessor, the Classic Query Engine (CQE). SQE is the strategic database engine, thus will receive all future enhancements. Although CQE will still be supported and maintained, IBM will not incorporate any new features into this engine. Only pure SQL requests are processed by the new engine. Because Query/400 is not an SQL interface, its query requests must be handled by the older CQE technology.

For more information about SQE including its advantages over CQE, see the *Preparing for and Tuning the SQL Query Engine on DB2 for i5/OS*, SG24-6598.

Over the years, thousands or perhaps millions of custom reports have been created using IBM Query for System i (also known as Query/400). As a result, businesses have a significant amount of investment contained throughout their systems in the query objects. Because so much has been invested in these objects, System i businesses need a way to leverage or repurpose them using a more modern reporting tool.

1.2 Introduction to DB2 Web Query for System i

In April 2007, IBM and Information Builders Incorporated announced a product alliance. The new product, called IBM DB2 Web Query for System i (DB2 Web Query), is the IBM strategic replacement product for Query/400. Like Query/400, it provides reporting capabilities against DB2 for i5/OS data. It also offers features that address the Query/400 limitations; it includes a graphical interface, modernized report formats, SQL access methods, and more. It can also run existing Query/400 reports, thus protecting the company's valuable investment in those system objects.

The DB2 Web Query product is available from IBM under licensed program 5733-QU2. Customers who have a license for Query/400, 5722-QU1, and are current on their software maintenance, are automatically entitled to the base license for 5733-QU2. Customers will be provided with a license for a limited number of named users based on software tier, unlike Query/400, which was licensed for unlimited users.

Query/400 and DB2 Web Query: In an attempt to reduce the naming ambiguity, throughout this book, the product Query for System i will be referred to as *Query/400* and the new product IBM DB2 Web Query for System i will be referred to as *DB2 Web Query*.

1.3 Synonyms

DB2 Web Query employs a data description language (DDL) that describes the necessary data elements for many types of data sources, including relational structures. Using this language, DB2 Web Query can describe i5/OS tables (physical files) or views (logical files) that have been established within the relational database system on the System i platform. This view of the data is a layer of abstraction that is used by DB2 Web Query for reporting and is known as *synonyms* or *metadata*. You can learn more about metadata in 3.4.1, "Creating metadata" on page 34.

Metadata and synonyms: The terms *metadata* and *synonyms* are used interchangeably throughout this book. They both refer to the DB2 Web Query data abstraction layer.

1.4 DB2 Web Query features

Licensed program 5733-QU2 has a base product and additional product components. We discuss the features that are supported by each of these options in the following sections.

1.4.1 Base features

In the following list, we highlight some of the main features of DB2 Web Query that are available with the base product of 5733-QU2:

- ▶ Browser-based interface

A Web browser is the interface for both DB2 Web Query developers and those users who will run the reports. This interface provides a graphical environment without the requirement of installing additional desktop software or plug-ins on user PCs.

- ▶ Query/400 report modernization

DB2 Web Query provides built-in adapters to import existing *QRYDFN objects without requiring any changes to those objects. After the appropriate metadata is created from the import process, the queries can be invoked from a browser interface and results displayed to the browser. This provides a modernized interface for running both existing queries and the report output.

- ▶ Report Assistant: Web-based query and reporting of data stored in DB2 for i5/OS

Included in DB2 Web Query is a component known as *Report Assistant*. This HTML-based graphical tool allows you to create new queries or modify existing ones, including imported Query/400 queries. You can select a data source, specify any sorting or grouping information, and display the report in your browser or another desktop application. When you use Report Assistant, DB2 Web Query creates a styled report that you can deploy on the Web without the necessity of learning the complexities of any reporting language.

- ▶ Delivery of output content in various modernized formats for Web browsers and wireless devices

By default, DB2 Web Query generates HTML and displays the query results to the browser. However, you can optionally specify other output formats. The following list shows the more commonly used formats that are available:

- Excel (97 and 2000)
- PDF
- XML
- PostScript®

In addition, you can direct your DB2 Web Query report output to the traditional spooled file and database file formats.

- ▶ Graph Assistant: Easy report graph generation

Graph Assistant is an HTML-based tool that guides you step by step through the creation of a graph. The Graph Assistant enables you to create and style your graphs.

- ▶ Power Painter: What You See Is What You Get (WYSIWYG) report building

Power Painter is a Web layout and report creation tool that is built by using Asynchronous JavaScript™ and XML (AJAX) technology. This tool enables you to create output and page layout formats. It combines reporting, graphs, and page layout design in a single tool. The framework makes Power Painter a thin-client tool that looks, responds, and performs like a Microsoft Windows®-based facility.

- Data adapters

Data adapters manage the communication between DB2 Web Query and the DB2 for i5/OS database through the use of SQL statements or other data access methods. Three adapters are provided with the base product:

- DB2 CLI
- DB2 Heritage files
- Query/400

For more information about each of these adapters, refer to 1.5.5, “Data adapters” on page 13.

- Runs natively on the System i platform

Each server side component of the DB2 Web Query solution runs natively on the System i platform. This integrated environment equates to a more simplified environment; there are fewer servers and operating systems to set up, administer, secure, backup, and maintain.

- Ability to use matrix reporting, ranking, color coding, drill-down and font customization to enhance the visualization of DB2 data
- Federated DB2 for i5/OS data access

Customers with multiple System i machines or i5/OS partitions will value the federated query capabilities of DB2 Web Query. This feature provides the ability to develop a single report against multiple DB2 for i5/OS databases.

For more information about the base component of 5733-QU2, see 2.1, “Installation and setup” on page 18.

1.4.2 Optional features available from IBM

In addition to the base product, optional features (also known as *add-ons*) are available to enhance the product’s functionality. Add-ons are available from IBM for an additional charge. The following features are included in the list of add-ons:

- Enablement of existing reports to use Online Analytical Processing (OLAP) or multi-dimensional functionality

The DB2 Web Query OLAP module allows slicing and dicing of data in an almost unlimited number of ways. This satisfies a broad range of analytical needs, enhances reports with extensive data-analysis functionality, and gives users the ability to dynamically interact with the data. OLAP enablement of a report allows users to drill down to finer levels of detail. Users can interact with their data, pivot rows and columns, visualize their numeric fields, and much more.

- Developer Workbench: Rich client for building customized business intelligence applications

DB2 Web Query Developer Workbench is an add-on component that provides a rich client development environment.

While you do not need Developer Workbench to create DB2 Web Query reports, there are compelling reasons to consider using it as your report development tool. The following list includes reasons to consider such an upgrade:

- Dimension tool
 - Defines hierarchies to be stored in the metadata
 - Is a prerequisite for OLAP module

- Interface to easily manipulate metadata and customize synonyms
 - Includes meaningful field names
 - Includes multilingual titles
- SQL Wizard that allows the developer to type or copy and paste SQL statements directly into the tool
- HTML Layout Painter tool

The powerful HTML Layout Painter tool allows you to combine reports, graphs, and other elements together in a single HTML document or layout. It provides a WYSIWYG design environment to help you create professional looking layouts that require more advanced components such as custom HTML, HTML template files, and cascading style sheets (CSS).

- Advanced programmer tools including:
 - Data profiling
 - Impact analysis
 - Ability to change field types (including integer or alpha to recognized date fields)
 - Automatic date decomposition
- Active Reports: Offline reporting capability

Many mobile workers require access to important information that is stored in their analytic reports while disconnected from the Internet. With DB2 Web Query Active Reports, such users can take the power of business intelligence with them wherever they go. By combining data and interactive controls into a single, self-contained HTML file, Active Reports delivers analytic capabilities in a completely portable and disconnected environment, with absolutely no software required. For more information about Active Reports, refer to Chapter 8, “Active Reports” on page 175, and refer to the following Web address:

http://www.informationbuilders.com/products/webfocus/portable_analytics.html

- Additional named user licenses

If you need more named users than the base option provides or you need more than you are currently licensed for, you can purchase additional licenses. For more details, contact your IBM representative.

1.4.3 Optional features available from Information Builders

In addition to add-ons that can be acquired from IBM, Information Builders offers a suite of products that can greatly enrich your ability to create and deliver high quality reports with a broad range of capabilities. The following more common add-ons can be acquired directly from Information Builders:

- Data adapters to connect to specific databases or enterprise resource planning (ERP) systems

DB2 Web Query provides comprehensive, native access to more than 300 distinct databases and information sources, including relational, legacy, ERP, customer relationship management (CRM), data staged in warehouses or data marts, and real-time data from operational systems, on any platform. You can find a current list of the native data, application, and technology adapters on the Web at:

<http://www.informationbuilders.com/products/systemi>

► WebFocus ReportCaster

ReportCaster is a scheduling and distribution application that centralizes the execution and distribution of WebFOCUS reports, the contents of URLs, and files. In addition to batch scheduling and execution of reports, the output can then be sent to users by e-mail, sent to a specific PC directory using File Transfer Protocol (FTP), or stored on the System i platform.

► Quick Data

Quick Data is an add-on for Excel that provides an Excel user with almost exactly the same interface as Report Assistant from within Excel.

Use Table 1-1 as a quick reference for determining which features are part of which offering.

Table 1-1 Feature offering matrix

Feature	Base offering	IBM add-on offering	Information Builder add-on offering
Base named user licenses	X		
Web browser interface	X		
Synonym creation	X		
Report Assistant	X		
Graph Assistant	X		
Power Painter	X		
Modernized output (HTML, XLS, PDF, and so on)	X		
Query/400 adapter	X		
DB2 Heritage File adapter (multi-member or multi-format files)	X		
DB2 CLI adapter (System i tables, physical files, views)	X		
DB2 for i5/OS cross-system join	X		
OLAP		X	
Developer Workbench		X	
Active Reports		X	
Additional named user licenses		X	
Report Caster			X
Over 300 additional data adapters			X
Quick Data Excel add-in			X

1.5 Architecture

Although DB2 Web Query runs natively on the System i platform, it is comprised of multiple application tiers that service different components. The overall architecture contains each of the following components:

- ▶ HTTP clients
 - Application server
 - Web server
- ▶ Reporting Server
- ▶ Data adapters
- ▶ Relational Database Management System (RDBMS) data

DB2 for i5/OS is the integrated database for the System i platform. Other source databases can be used with the installation of the appropriate add-on adapters that are discussed earlier in this chapter.

Figure 1-1 shows the overall architecture of the DB2 Web Query system.

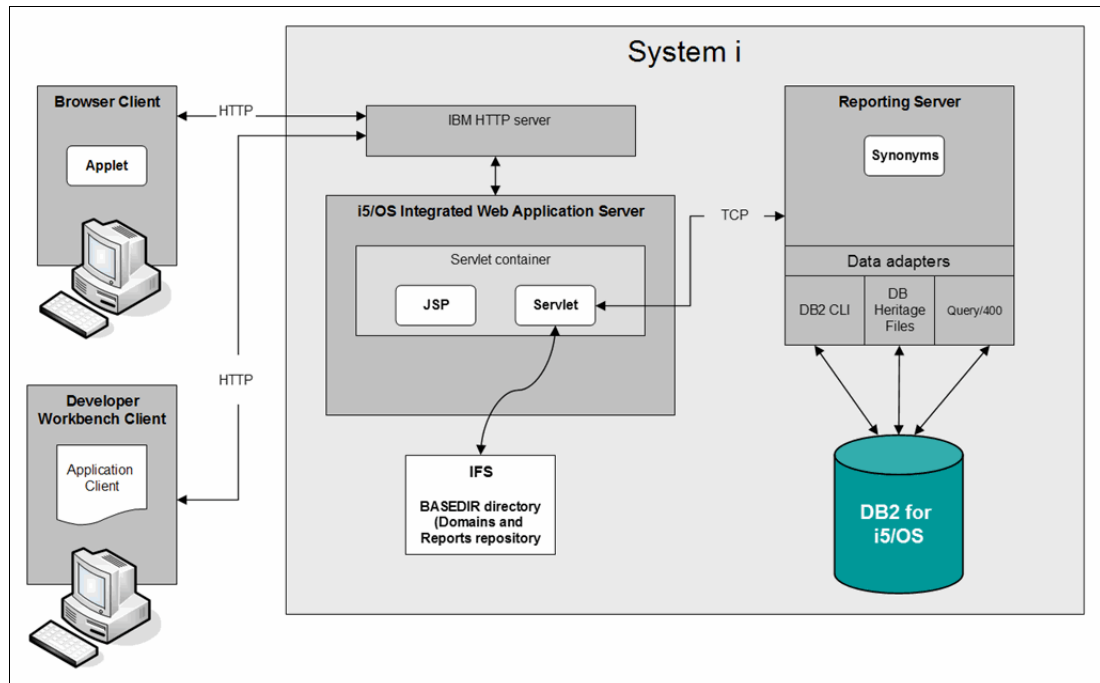


Figure 1-1 DB2 Web Query base architecture

To better understand the architecture, it is helpful to understand the end-to-end process and how a report request flows through the architecture. The following actions describe the life cycle of a typical DB2 Web Query request:

1. A user requests a report for execution from a Web browser.
2. The Web server receives the request, processes the parameters, and routes it to the Reporting Server via the DB2 Web Query servlet.
3. The Reporting Server processes the request and passes it to the appropriate data adapter.

4. The data adapter generates the appropriate database request and submits the request to the DB2 for i5/OS database engine.
5. The Reporting Server receives the result set from the database, formats the report, and returns the formatted report to the Web server via the DB2 Web Query servlet.
6. The Web server delivers the report to the user's Web browser for display.

1.5.1 Web browser clients

Both DB2 Web Query developers and users use a Web browser as the interface to the product. The browser clients communicate via HTTP to the application server. The following Web browsers are supported:

- ▶ Internet Explorer® (IE) version 6.0 or later
- ▶ Mozilla Firefox version 1.5 or later

1.5.2 Web server

Web servers generally serve as the static content repository and handle HTML, GIF, CGI, and other traditional Web content and processing. The IBM HTTP Server is used to fulfill this role for the DB2 Web Query product.

The Web server content for DB2 Web Query is grouped into contexts (aliases). These are grouped together in subdirectories and stored in the i5/OS integrated file system under the qibm/proddata/webquery/ibi/webfocus76 directory. Under this directory is webquery_html, the context subdirectory that contains the following items:

- ▶ Common templates and forms
- ▶ Common images, scripts and style sheets
- ▶ Published forms and launch pages

1.5.3 Application server

An *application server* is a component-based product that resides in the middle-tier of a server-centric architecture. It handles requests from Web clients that require Java and non-traditional processing and provides middleware services such as security and state maintenance. DB2 Web Query runs on the i5/OS Integrated Web Application Server and uses the servlet container component to process these client requests. Servlet containers (also referred to as *servlet engines*).

The DB2 Web Query application server has the following attributes:

- ▶ Is a J2EE™ compliant Web application
- ▶ Runs on a *servlet container*, a Java-based container that serves servlets, JavaServer™ Pages™ (JSPs) and connectors
- ▶ Does not require a full J2EE application server
- ▶ Supports servlets
- ▶ Standardizes deployment schemes using archives or packages

- In some cases, groups multiple applications into Web Application Bundle (WAB) files
Applications are grouped into the WAB file named webfocus76.wab. This WAB file is served under the context root of /ibi_apps and contains the following components:
 - DB2 Web Query servlet
 - DB2 Web Query API
 - Managed Reporting API

Communicates with the MR Repository (basedir)

 - DB2 Web Query home page controller and JSPs
 - DB2 Web Query administration console
 - DB2 Web Query and Managed Reporting drivers
 - Communication Drivers to DB2 Web Query server
 - Custom DB2 Web Query exit (plug-in)
 - Custom servlet filter (plug-in)

Figure 1-2 shows the DB2 Web Query servlet within the base architecture.

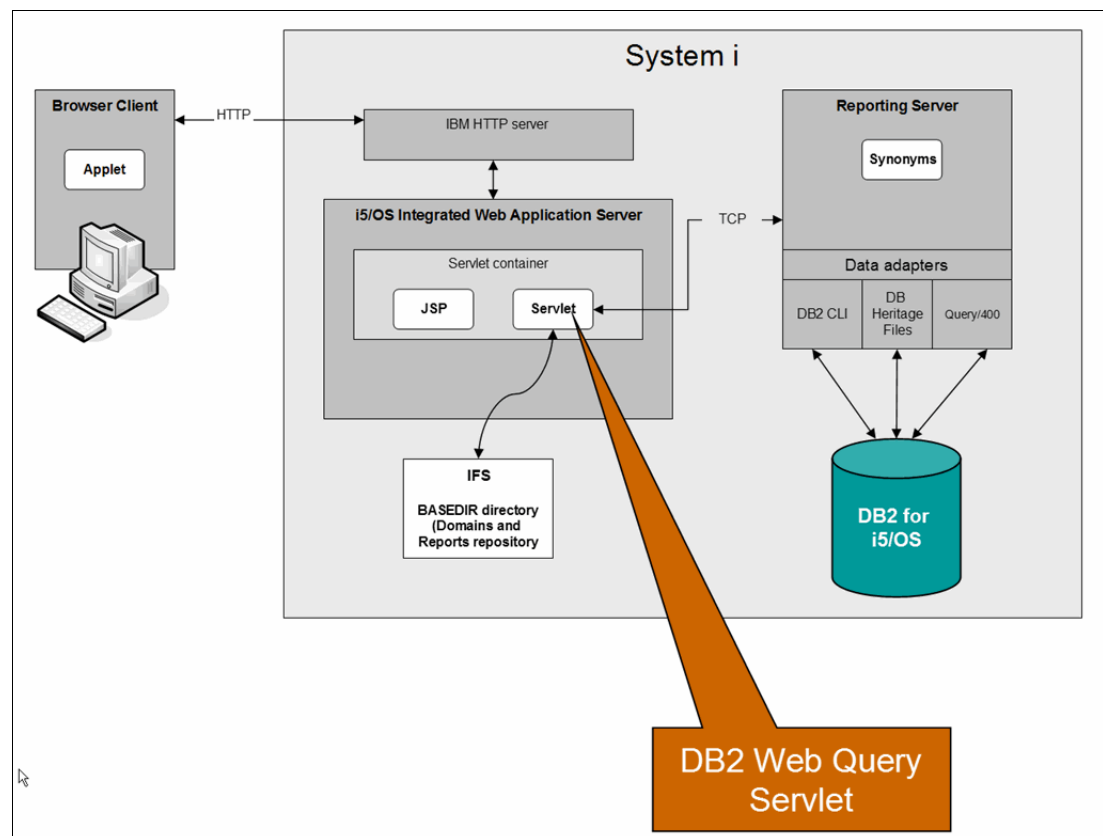


Figure 1-2 DB2 Web Query Servlet within the application server

1.5.4 Reporting Server

The DB2 Web Query Reporting Server is a C-based application that resides on the System i platform and is responsible for managing data access, processing the business logic, and generating the fully styled output. It is comprised of the following components:

- Reporting Engine of DB2 Web Query product
- Data adapters repository
- Metadata or synonym repository

1.5.5 Data adapters

An *adapter* is a program that enables DB2 Web Query to access a data source. Data adapters are responsible for generating the appropriate request to submit to the database engine. DB2 Web Query comes with three different adapters. The adapter that you choose to create your metadata or synonym is transparent to the users who will create reports. The format of the request depends on the adapter that is used:

- The *DB2 CLI adapter* generates appropriate SQL statements to submit to the DB2 for i5/OS database engine.

This is the recommended adapter for use by DB2 Web Query. Select this adapter if you want to write a report on a single member physical file. The report can include an SQL table or a data description specifications (DDS)-created physical file. This adapter is also used for object types of alias, stored procedures, and materialized query tables (MQTs). The DB2 CLI adapter generates SQL that takes full advantage of the latest DB2 enhancements that are found in the SQL Query Engine (SQE).

- The *DB2 Heritage File adapter* generates the appropriate OPNQRYF command to handle reports that are based on data in a file that has multiple members or multiple record formats.

This command does not take advantage of the new enhancements to the DB2 optimizer and uses the Classic Query Engine (CQE) under the covers.

- The *Query/400 adapter* retrieves the associated Query/400 query.

Choose this adapter when you want to create metadata on an already existing Query/400 query. The QRYDFN object must exist on the System i platform. This adapter sends the RUNQRY command to the server, which also uses the older CQE.

Table 1-2 summarizes the adapters and how they correspond to the data types.

Table 1-2 Data adapters

Adapter	Data type	Command sent to System i5™ machine
DB2 CLI	Single member physical file; alias, stored procedure, or MQT	CLI API
DB2 Heritage File	Multiple member file, multi-record format files	OPNQRYF
Query/400	QRYDFN object	RUNQRY

The three adapters are provided with the base DB2 Web Query product. To see a full list of additional adapters that can be purchased from Information Builders, refer to the following Web address:

http://www.informationbuilders.com/products/webfocus/data_access.html

1.6 DB2 Web Query Developer Workbench

Developer Workbench serves as the “thick” client component of DB2 Web Query that is used primarily for advanced development purposes. It is the standard integrated development environment (IDE) for DB2 Web Query technology. Similar to the browser client, the Developer Workbench client communicates in HTTP to the DB2 Web Query application server. Developer Workbench is an optional feature of DB2 Web Query. Developer Workbench is the only component that requires any software to be loaded on the client. Figure 1-3 shows its place in the overall architecture.

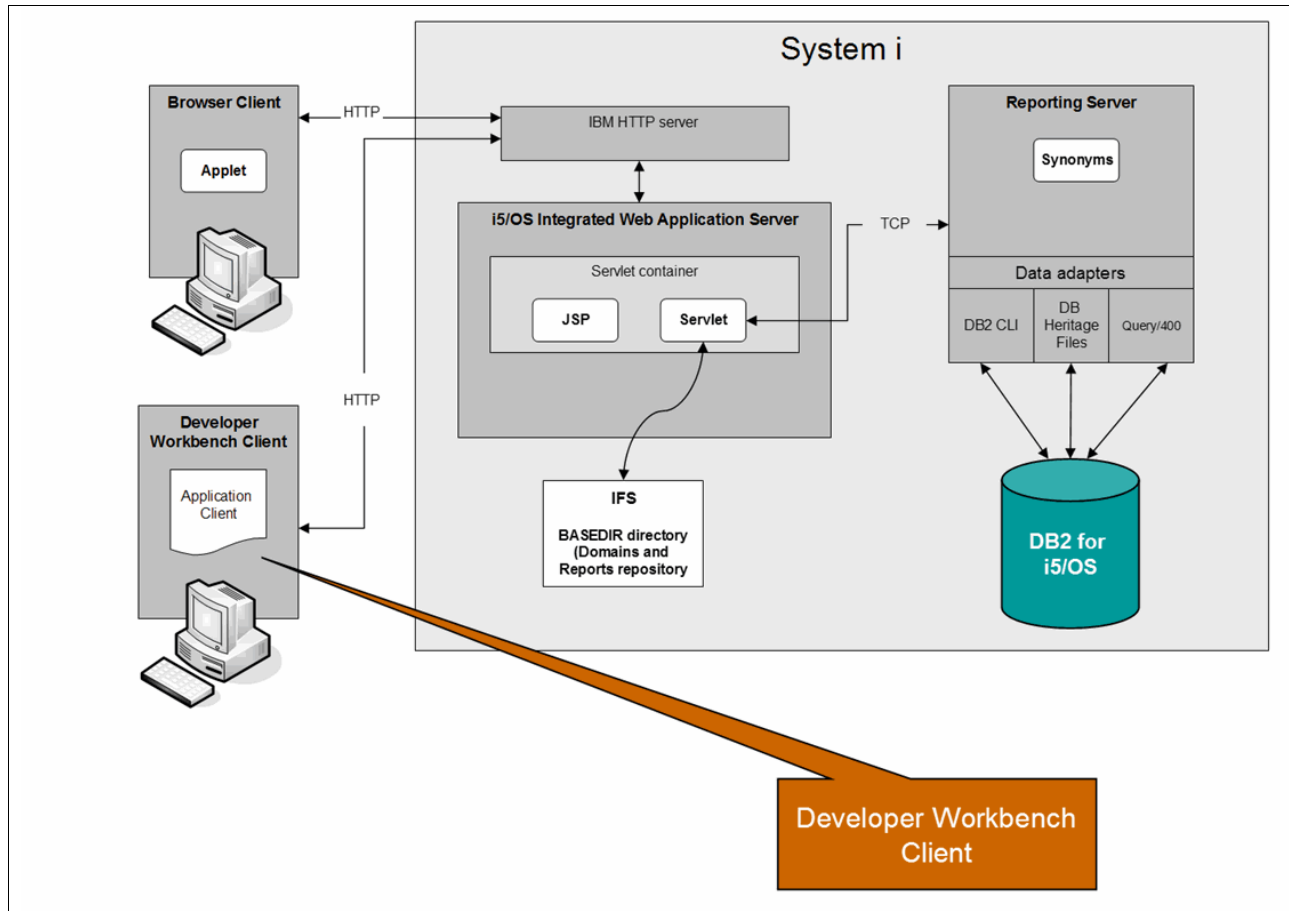


Figure 1-3 Developer Workbench within the DB2 Web Query architecture

1.7 DB2 Web Query product positioning

The base version of the DB2 Web Query product is intended to serve as the strategic replacement to Query/400. If you are using Query/400 today as your primary reporting tool, or your requirements are for a robust query and reporting tool on the System i platform, the base product is all that you need. While the new Web-based product is ideal for the mid-market, it offers the foundation to support large enterprise requirements as well. Consequently, if you have requirements for more advanced business intelligence capabilities, the upgrade path (available through the various add-on components) allows DB2 Web Query to be well positioned to support such large enterprise needs as well. This product fills user reporting requirements from parameterized and ad hoc reporting, to analytical (multidimensional)

slicing and dicing. It does not require a data warehouse or cubing technology, but if that is the desired business intelligence infrastructure, it integrates with that design with ease.

The add-on components that are available from Information Builder provide the ability to integrate with over 280 sources of information, including legacy sources and packaged applications. This is accomplished through native data adapters that are tuned and optimized for each data source. Such integration can greatly enhance the potential of a data warehouse.

One of the biggest reasons why a data warehouse implementation fails is because it lacks certain key information to make it complete. This is often because the source for the missing data was inaccessible by the warehousing tool; this problem is addressed by DB2 Web Query. For example, in a single report, you can view data that came from an Oracle database, an ERP system, and a DB2 data mart or data warehouse repository.

Overall you will find that the various data adapters can reduce the data integration costs for your business intelligence implementations because of its simple access to these sources. Reports from these various data sources can be created within hours after product installation.

Because DB2 Web Query is comprehensive and can access all data in a company, it is the logical choice as the standard for business intelligence and reporting for many companies. You can use it for management within your organization to provide the following types of reports to keep you updated on the general health of your business:

- ▶ Dashboards
- ▶ Scorecards
- ▶ Key performance indicators (KPI)
- ▶ Drill downs
- ▶ Operational reports

In addition, the ReportCaster product that is available from Information Builders gives you the ability to automatically deliver reports to your users. This add-on component can intelligently burst the pages of a single report to the appropriate people. Delivery can be based on a schedule or technology event, such as a new transaction, the rise or fall of a stock price to a certain level, or even a change in the weather. Delivery destinations include sending reports to a Web server, a database archive (called a *Report Library*), to e-mail (one of the more common destinations), mobile devices, and to a printer or fax machine.



Installation and server operations

In this chapter, we explain the installation of DB2 Web Query, 5733-QU2, using the RSTLCPGM command. We outline the PC and System i prerequisites for installing DB2 Web Query. We also describe which i5/OS system jobs will run and their functions in DB2 Web Query. In addition, we discuss other system objects that are created upon installation of DB2 Web Query.

Note: For the most current and complete information about the installation instructions, see the *IBM DB2 Web Query for System i V1R1M0 Install Instructions* on the Web at:

http://www-03.ibm.com/systems/i/software/db2/pdf/DB2_Web_Query_Install.pdf

2.1 Installation and setup

The licensed program product number for DB2 Web Query for System i is 5733-QU2, which replaces 5722-QU1 (Query/400). DB2 Web Query includes the existing Query/400 product as well as the new offering. You can continue to create and run Query/400 reports and slowly migrate to DB2 Web Query if you choose. The two products can successfully operate concurrently against the same tables and libraries.

Query/400, product number 5722-QU1, as an independent product, will be withdrawn from marketing. The functionality will reside under 5733-QU2 with DB2 Web Query.

2.1.1 Installing DB2 Web Query

Important: Before you start the installation, you must obtain the most current information about the prerequisites and required program temporary fixes (PTFs). You can find this information in Informational APAR II14318, which is available on the Web at:

http://www-912.ibm.com/n_dir/nas4apar.nsf/c79815e083182fec862564c00079d117/19f34f5766e54eef8625732d003c677d?OpenDocument&Highlight=2,II14318

After you install the required license programs and PTFs found in the APAR, follow these steps to install DB2 Web Query:

1. Sign on as QSECOFR or use a user profile with *SECADM and *ALLOBJ authority.
2. For the Base product, enter the following command:

```
RSTLICPGM LICPGM(5733QU2) DEV(OPT01)
```
3. Depending on the additional options of the product that you want to install, type the following commands:
 - If you purchased Active Reports, enter:

```
RSTLICPGM LICPGM(5733QU2) DEV(OPT01) OPTION(1)
```
 - If you purchased the OLAP Enablement option, enter:

```
RSTLICPGM LICPGM(5733QU2) DEV(OPT01) OPTION(2)
```
 - If you purchased the Developer Workbench, enter:

```
RSTLICPGM LICPGM(5733QU2) DEV(OPT01) OPTION(3)
```
4. Read the license agreement and press F14 to accept it.
5. Load and apply the latest 5733-QU2 PTFs as explained in Informational APAR II14318 (see the Important notice at the beginning of this procedure).
6. Register the DB2 Web Query HTTP instance by typing the following command:

```
CALL QWEBQRY76/REGWEBQRY
```
7. The first time the product is installed, it creates a user profile QWEBQRYADM with no password supplied. Set up a password for this profile by typing the following command:

```
CHGUSRPRF USRPRF(QWEBQRYADM) PASSWORD(yourpassword)
```

Developer Workbench: If you install Developer Workbench, after you complete the previous steps, complete these steps:

1. Load and apply the latest 5733-QU2 PTFs for Developer Workbench as explained in Informational APAR II14318.
2. Copy the installed file from integrated file system to your PC. You can do this by using either File Transfer Protocol (FTP) on the install file to the PC or by mapping a drive from the PC to the integrated file system. The installation file is located in the /qibm/proddata/webquery/ibi/DeveloperWorkBench/WfDevStudio763i.exe directory.
3. Execute the Developer Workbench Install Program, WfDevStudio763i.exe.

i5/OS system objects

Upon installing 5733-QU2, the following integrated file system directories are created:

- ▶ /QIBM/PRODDATA/WEBQUERY/IBI
- ▶ /QIBM/USERDATA/WEBQUERY/IBI

Table 2-1 shows the system objects that are created.

Table 2-1 System objects

Object name	Object type
QWEBQRY76	*LIB
QWEBBASE	*LIB
QSYS/QWEBQRYADM	*USRPRF
QSYS/MRADMIN	*USRPRF
QSYS/MDUNTITLED	*USRPRF
QSYS/QWEBQRY	*AUTL
QWEBQRY76/QWEBQRYJOB	*JOB
QUSRSYS/QWEBQRYADM	MSGQ

These are DB2 Web Query-related user profiles:

- ▶ QWEBQRYADM: This profile has functionality on the server side and has authority to start and stop the Reporting Server. The profile is automatically created.
- ▶ MRADMIN: This group profile has the authority to create domains in DB2 Web Query. It also has the rights to create reports in all domains and folders on the system.
- ▶ MDUNTITLED: This group profile gives users the authority to create reports in the default common domain.

2.1.2 Authorizing and verifying users

The users must be authorized to use DB2 Web Query. For each user that will use the product, type the following command:

```
CALL QWEBQRY76/WQADDLIC 'USRPRF'
```

Here USRPRF is the user profile that you want to authorize. Type the USRPRF name in uppercase.

If you need to remove a user, type the following command:

```
CALL QWEBQRY76/WQLSLIC 'USRPRF'
```

Here USRPRF is the user profile that you want to disable.

Note: If you exceed the limit of purchased named users, the login function is disabled.

To display which users are currently registered, type the following command:

```
WRKLICINF PRDID(5733QU2)
```

Then on the next panel, select option 8 for feature 5050.

2.1.3 License keys

After you install DB2 Web Query, you are allowed a grace period in which to try the product. During this grace period, you have access to all the features that are provided in the base product as well as all of the IBM add-on components. However, after the grace period expires, license keys are required to continue using both the base product and the add-on components.

To obtain your DB2 Web Query license keys, you need to work through your normal product ordering channels (for example, a System i business partner). The system serial number is required to generate your specific license keys. When you receive this information back from your business partner, it looks like the example shown in Figure 2-1 on page 21.

Each license key is used to activate either the base product feature (5050) or one of the add on features (5101 and 5102). To enter your license keys and activate the specific features, use the ADDLICENSE command. For the examples given in Figure 2-1 on page 21, you enter the following commands:

```
ADDLICENSE PRDID(5733QU2) LICTRM(V1R1M0) FEATURE(5050) SERIAL(6504690)  
LICENSE(921BA3 71228D 4F7C01 ) USGLMT(4) EXPIRE(*NONE)
```

```
ADDLICENSE PRDID(5733QU2) LICTRM(V1R1M0) FEATURE(5101) SERIAL(6504690)  
LICENSE(844C88 0793CB 21B986 ) USGLMT(4) EXPIRE(*NONE)
```

```
ADDLICENSE PRDID(5733QU2) LICTRM(V1R1M0) FEATURE(5102) SERIAL(6504690)  
LICENSE(896174 D047B4 548C01 ) USGLMT(4) EXPIRE(*NONE)
```

Note: You do not have to wait for the grace period to expire before you enter your license keys. You can do this at any time.

Product ID	: 5733QU2
License term	: V1R1M0
Feature	: 5050
Serial number	: 1078950
Processor group	: *ANY
Expiration date	: 07/11/30
Usage limit	: 4
Vendor data	: *NONE
License key	: 921BA3 71228D 4F7C01
Product ID	: 5733QU2
License term	: V1R1M0
Feature	: 5101
Serial number	: 1078950
Processor group	: *ANY
Expiration date	: 07/11/30
Usage limit	: 4
Vendor data	: *NONE
License key	: 844C88 0793CB 21B986
Product ID	: 5733QU2
License term	: V1R1M0
Feature	: 5102
Serial number	: 1078950
Processor group	: *ANY
Expiration date	: 07/11/30
Usage limit	: 4
Vendor data	: *NONE
License key	: 896174 D047B4 548C01

Figure 2-1 DB2 Web Query license keys

2.2 Requirements

In this section, we cover the PC and System i minimum requirements to install and run DB2 Web Query.

2.2.1 PC requirements

No client is required to be installed on your PC to enable DB2 Web Query; you only need a browser. The browser requirements are:

- ▶ Internet Explorer 6.0 or later
- ▶ Firefox 1.5 (certified), version 2.0 (supported)

If you are unsure about the version of Internet Explorer that you are using, open the browser and select **Help** → **About Internet Explorer**. Note the version number.

In regard to PC memory requirements, in general, queries usually run best on PCs that have 1 GB of RAM or more.

2.2.2 System i requirements

Your System i environment must meet the following requirements:

- ▶ V5R4 of i5/OS is required.
- ▶ Qshell must be installed. It is a no charge option and comes with i5/OS.
If you are unsure about whether it is installed, type QSH on your System i5 command line and press Enter. If a QSH command line appears, then it is installed. It is option 30 of the operating system.
- ▶ The following licensed programs must be installed:
 - 5722SS1 option 30 - Qshell
 - 5722SS1 option 33 - Portable App Solutions Environment
 - 5722JC1 - IBM Toolbox for Java
 - 5722DG1 - IBM HTTP Server for i5/OS
 - 5722JV1 option 7 - Java Developer Kit 5.0
 - 5722JV1 option 8 - J2SE™ 5.0 32 bit
 - 5722JV1 - Java Runtime Environment (JRE™) 1.4 or later

2.2.3 Developer Workbench requirements

Although Developer Workbench client is not required for DB2 Web Query report development and execution, you might still find that it provides additional development features that make it worthwhile. If you choose to install it on your PC, use the following hardware and software requirements.

Hardware requirements

Verify that your Windows machine meets the Developer Workbench hardware requirements:

- ▶ Pentium® 3 or later at 600 MHz or later
- ▶ VGA or later resolution graphics card; display of 1027x768 recommended
- ▶ 512 MB of RAM
- ▶ 500 MB free hard disk space for CD installation; 1.5 GB for downloaded installation

Software requirements

Verify that your Windows machine meets the Developer Workbench software requirements:

- ▶ Windows 2000, XP, or 2003 Server and Professional Editions
Windows 2000 must have Service Pack 2 or later
- ▶ Internet Explorer 6 and later
Internet Explorer components are required for Developer Workbench. If possible, upgrade to the most recent version of Internet Explorer.
- ▶ Adobe® Acrobat® Reader 6.0 and later.
Acrobat Reader 6 and 7 are certified to view PDF reports generated by Developer Workbench.

Note: You must be an administrator to the Windows machine to run the Developer Workbench installation.

Provided third-party component

The third-party component Java 2 SDK 1.5.0_09 is provided for use with Developer Workbench. A Java SDK is required for WebFOCUS features such as servlet connectivity, graph generation, and online analytical processing (OLAP). If it is not present on your machine, you have the option to install it with Developer Workbench.

If Java SDK 1.4.1 or later is not installed on your machine, you *must* allow Developer Workbench to install the SDK.

2.3 Reporting Server commands

The base component of the licensed program provides multiple i5/OS commands to help with DB2 Web Query administrative tasks. In order for these command to work successfully, you must be signed in as user profile QWEBQRYADM.

- ▶ **STRWEBQRY START(*ALL)**

Use the STRWEBQRY command to start all the jobs that are required to run Web Query.

- ▶ **ENDWEBQRY**

Use the ENDWEBQRY command to end the Reporting Server. We recommend that you use the OPTION (*CLEAR).

- ▶ **TRCWEBQRY**

The TRCWEBQRY command turns on, turns off, and saves server traces.

2.4 DB2 Web Query server jobs

DB2 Web Query connects to i5/OS using the SQL *call-level interface* (CLI). CLI is a callable SQL programming interface that is available in DB2 on i5/OS. CLI consists of application programming interfaces (APIs) that are used to connect to the server and start dynamic SQL statements. The CLI is a subset of Open Database Connectivity (ODBC). The system job that processes CLI is QSQSRVR. For more information about CLI, refer to *System i Database DB2 UDB SQL call level interface (ODBC)* on the Web at:

<http://publib.boulder.ibm.com/infocenter/iseres/v5r4/topic/cli/rzadp.pdf>

The following jobs run on your server when the Reporting Server is running. For all jobs, the start time of the job is the start time of the Reporting Server.

- ▶ **EDAPTH:** There is one job that helps with the workspace process.
- ▶ **EDAPLOG:** There is one job that contains startup information.
- ▶ **EDAPGWY:** There are three of these jobs, which are listener jobs, one each for HTTP, TCP, and Java. These jobs receive incoming requests and hand off work to the TSCOM3 jobs.
- ▶ **TSCOM3:** By default, four of these jobs are running. These jobs accept the request from the EDAPGWY job and translate the DB2 Web Query request into SQL. One TSCOM3 job does not correspond to one user; one TSCOM3 job can service several users. For this reason, the design of these jobs contributes to the scalability of the product. They are referred to as “agents.”
- ▶ **JSCOM3:** There is one job, which services Java processes.
- ▶ **HLISNK:** There is one job, which is an internal server process.

- ▶ QP0ZSPWP: This job is the DB2 Web Query JVM™ thread.
- ▶ WQLWI7: This job is for integrated application server jobs. Three of these jobs should be running whenever DB2 Web Query is active.
- ▶ QSQSRVR: This i5/OS native prestart job handles SQL requests that are made over CLI. This job does the actual database work of optimization and returning the result.

All these jobs are started in QSYSWRK when the STRWEBQRY command is used. The QSQSRVR jobs always run in the QSYSWRK subsystem, even if you configure all the other jobs to run in their own subsystem.

You can find information about the runtime environment, including the ports that are used, in `/QIBM/UserData/webquery/ibi/srv76/wfs/edaprint.log`.



DB2 Web Query fundamentals

In this chapter, we introduce the basic steps for you to start using DB2 Web Query. In 3.2, “Logging in” on page 26, we illustrate how to sign on to DB2 Web Query. After you log on, proceed to the respective section that contains the instructions depending on the type of user you are: an administrator, a developer, or a basic user:

- ▶ 3.3, “Administrator tasks” on page 28
- ▶ 3.4, “Developer tasks” on page 32
- ▶ 3.5, “Basic user” on page 53

3.1 Registering named users

DB2 Web Query is licensed by named users. Each user profile needs to be registered.

Registering a new user

To register a user with DB2 Web Query, enter the following command:

```
CALL QWEBQRY76/WQADDLIC 'USERPROFILE'
```

USERPROFILE references your System i user profile name and must be specified in uppercase letters.

Removing a DB2 Web Query registration

To remove a registered DB2 Web Query user, enter the following command:

```
CALL QWEBQRY76/WQRLSLIC 'USERPROFILE'
```

Again your user profile must be in uppercase.

Displaying the registered users

To display your currently registered users, enter the following command:

```
WRKLICINF PRDID(5733QU2)
```

Then select option 8 for feature 5050.

3.2 Logging in

One of the many positive features of DB2 Web Query is that you do not need to install anything on your PC to use it. You only need a browser to create and run reports. As mentioned in 2.2.1, "PC requirements" on page 21, DB2 Web Query supports Internet Explorer and Firefox.

In this example, the system name is MYSYSTEM.ABC.ACME.COM:

1. Enter the following internal URL:

```
http://mysystem.abc.acme.com:11331/webquery
```

The port number in the URL is *always* 11331, since this is the default port for DB2 Web Query.

2. Sign onto DB2 Web Query with the same user profile and password that you use to access the System i machine. A Web page opens like the example shown in Figure 3-1.

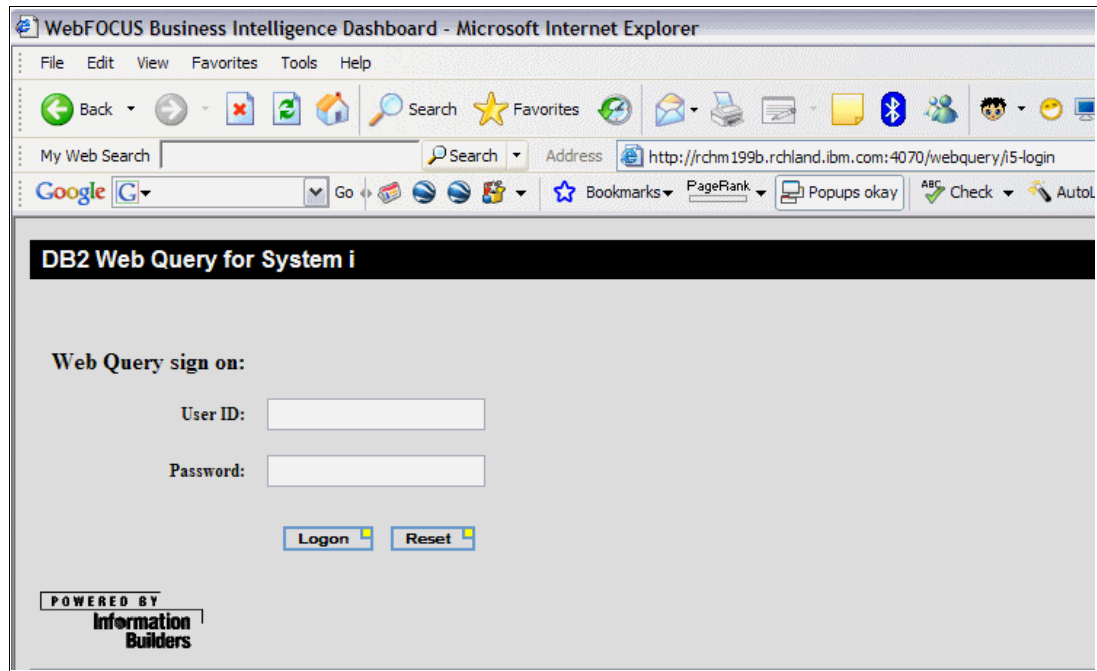


Figure 3-1 DB2 Web Query signon page

After you log on, you see only the domains and menu options that you are authorized to as shown in Figure 3-2. For an explanation about domains, refer to 3.3.1, "DB2 Web Query domains" on page 28. If you do not see a domain that you think you should be authorized to, contact your DB2 Web Query administrator.

Tip: Add your DB2 Web Query URL to your browser "Favorites".

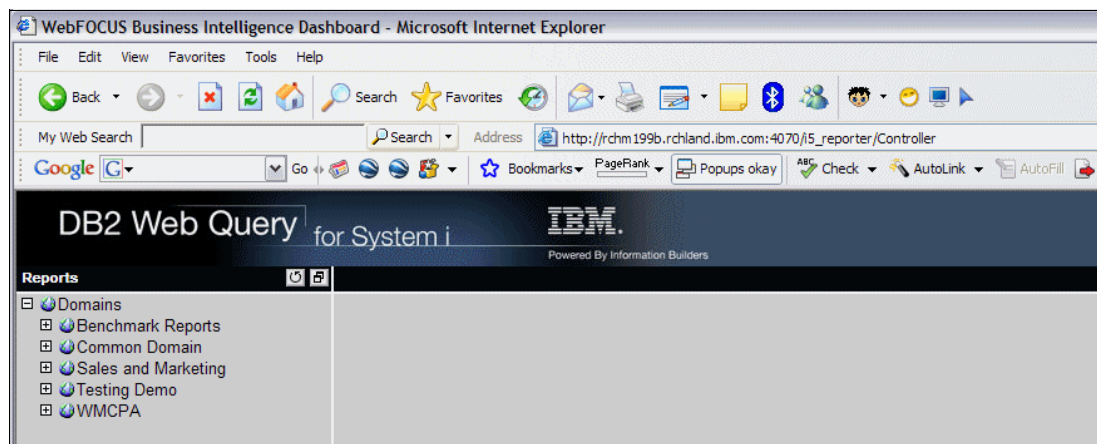


Figure 3-2 DB2 Web Query home page

3.3 Administrator tasks

The main tasks for administrators are to create domains, create subfolders within those domains, and assign user authority.

To perform the administrator tasks, the administrator must have a user profile. The group profile for the administrator is MRADMIN. This administrator is the only one who is authorized to create new domains. The administrator is authorized to create and run reports in all domains.

If user profile SAMMY is to be the administrator, use the following CL command to assign it to the group profile MRADMIN:

```
CHGUSRPRF USRPRF(SAMMY) GRPPRF(MRADMIN)
```

If the user profile SAMMY already has a group profile, then use the following command so that you do *not* to overwrite the current group profile:

```
CHGUSRPRF USRPRF(SAMMY) SUPGRPPRF(MRADMIN)
```

3.3.1 DB2 Web Query domains

In the context of DB2 Web Query, the term *domain* refers to a graphical way to classify and separate your reports. For example, an administrator can create domains for Sales, HR, Marketing, and so on.

The first time you open your browser to DB2 Web Query, you see only the default Common Domain folder. Inside the Common Domain folder, you see two other folders as shown in Figure 3-3:

- ▶ The *Reports folder* is where permanent report objects are displayed. Users go into this folder to find the report they want to run. Developers that are authorized to this domain can create new reports in this folder.
- ▶ The *Other Files folder* contains files like style sheets and JPG files that you might want to import into your report.



Figure 3-3 Common Domain folder

Naming your domain

It is good practice to initially name your domains with exactly eight characters. These eight characters are also known as the *Href*. We explain more about the eight-character rule in the following section.

To view an Href for a domain, as shown in Figure 3-4, right-click the domain and select **Properties**. If you create a domain that has less than eight characters, the domain name is padded to complete the eight characters. The padding characters are autogenerated. You need to know what the Href is in order to assign user authority, which is explained in 3.3.3, “Assigning authority to users” on page 31.

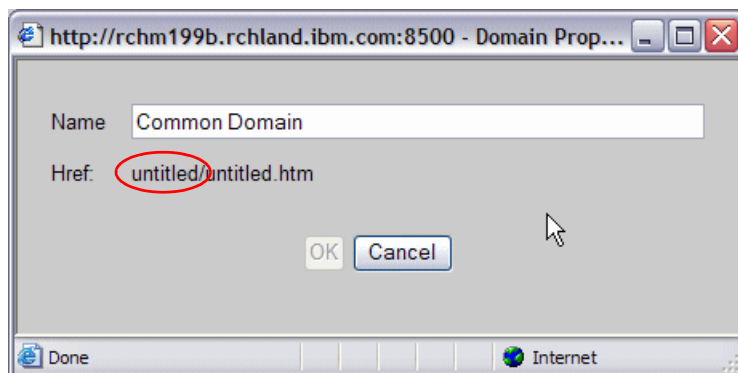


Figure 3-4 Properties of a domain

Reason for creating domains with exactly eight characters

DB2 Web Query uses the Href, which is based on the domain name, to control user access to the domain. DB2 Web Query checks which System i groups the user belongs to. If the user belongs to a group named “MDhrefname”, the user can be a developer in the domain with the associated Href. For a user to be a developer in the Common Domain (Figure 3-4), they must belong to the group MDUNTITLED.

The other special prefix is that DB2 Web Query checks for MU. Having a group profile of MUUNTITLED means that the profile is a valid user in the Common Domain (Href “untitled”) and is allowed to run reports there but not to create them. In fact, the Common Domain is the system default and is the one domain in which you do not require an MUhref to run reports. If you are registered as a licensed user of DB2 Web Query, then you are entitled to run reports in the Common Domain by default.

If you want a domain name to be greater than eight characters, first create it with a meaningful name of exactly eight characters. Then rename your domain to a desired name that is greater than eight characters. The original eight characters remain as your Href name.

For example, if you want your domain to be “Month End Reports”, you can first create your domain as “monthend”. Then you can rename it to Month End Reports and your Href remains as the eight characters of “monthend”. Table 3-1 shows the Developer and Basic User group profiles for our example.

Table 3-1 Developer and basic user group profiles

Domain name	Href name	Developer group profile	Basic user group profile
Month End Report	monthend	MDMONTHEND	MUMONTHEND

Creating a domain

To create a domain:

1. Log on to the DB2 Web Query home page as administrator, which is a user profile that belongs to the MRADMIN group.
2. Right-click **Domains** and select **New Domain**.

3. In the New Domain window that opens, like the one in Figure 3-5, type your domain name and click **Save**.

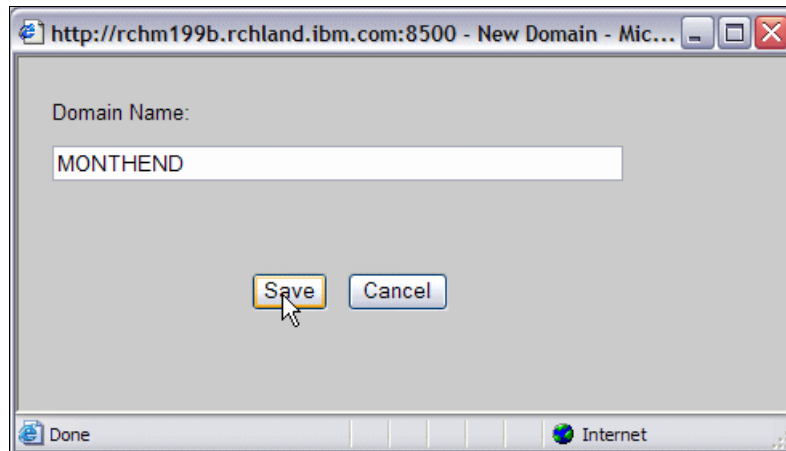


Figure 3-5 New Domain window

4. If your domain name was shorter than eight characters, right-click your domain and select **Properties** in order to view what was autogenerated.

Tip: Avoid using blanks or special characters in domain names, folder names, or field names within DB2 Web Query.

Renaming a domain

To rename a domain:

1. Right-click the existing domain and select **Properties**.
2. In the Domain Properties window (Figure 3-6) that opens, in the Name field, type a preferred name over the name in the white box. After you edit the name, click **OK**.

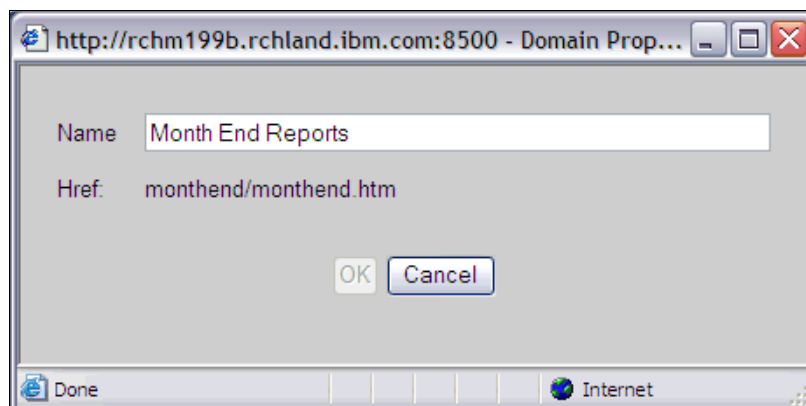


Figure 3-6 Domain Properties window

3.3.2 Creating subfolders

After you create your domains, you must create folders inside of those domains. These subfolders further classify the types of reports within their domains. When the domain is expanded, a developer or user can recognize which subfolder contains the desired report.

Keep in mind that you *cannot* create reports directly in the Reports folder. You must use a subfolder to create and store reports.

Only a developer is authorized to create new subfolders; a user is not.

To create a subfolder:

1. Right-click **Reports** and select **New Folder**.
2. In the New Folder window (Figure 3-7) that opens, type the desired name for the new folder and click the **Save** button.

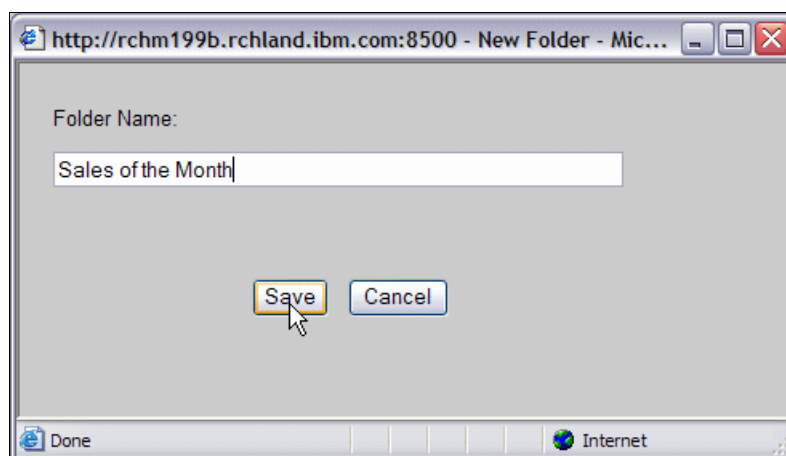


Figure 3-7 New Folder window

Now developers can create reports in the folder.

3.3.3 Assigning authority to users

There are two other user categories besides the administrator. The two categories are developers and users. Developers can create new reports and modify existing reports. Users only have authority to run existing reports. They cannot edit or change anything about the existing report. When basic users right-click a report, they do not see a menu.

Table 3-2 shows the user types and the authority that they have. MONTHEND is the example domain name that we use. MD denotes the developer group profile for that domain, and MU denotes the basic user group profile for that domain.

Table 3-2 Authority of the different users

	Administrator	Developer	Basic user
Group Profile name	MRADMIN	MDMONTHEND	MUMONTHEND
Authority			
Create domain	Yes		
Create subfolder	Yes	Yes	
Create report	Yes	Yes	
Change report	Yes	Yes	
Run report	Yes	Yes	Yes

Here is an example of creating a new domain and assigning user authority. Let us say that you create a domain named *monthend*. Bob is a developer, and Jim is a basic user who runs read-only reports. Follow these steps:

1. Create the new domain from the DB2 Web Query home page.
2. From the System i command line, create the user profiles using the following CL commands:


```
CRTUSRPRF USRPRF(MDMONTHEND) TEXT('Group Profile for Developers')  
CRTUSRPRF USRPRF(MUMONTHEND) TEXT('Group Profile for Basic Users')
```
3. After you create the group profiles for Developers and Users for the domain *monthend*, assign the existing users their correct authority. Enter the CHGUSRPRF CL command as follows:

```
CHGUSRPRF USRPRF(BOB) GRPPRF(MDMONTHEND)  
CHGUSRPRF USRPRF(JIM) GRPPRF(MUMONTHEND)
```

If you use the CHGUSRPRF command to change the group profile name while a user is currently signed on to DB2 Web Query, the user must log off and log on again to see the new authority reflected.

If a user does not belong to any group profiles for a domain, the user is only authorized to run reports in the Common Domain. No other domains are visible to the user.

3.4 Developer tasks

In this section, we describe the developer tasks for creating a DB2 Web Query report. We explain the steps that prepare you to write a report, such as which adapter to use and creating metadata.

A developer does not need to know either Query/400 or SQL to use DB2 Web Query. The report writing process in DB2 Web Query is different than either of the former tools. Although DB2 Web Query is mentioned as a replacement for Query/400, it is not a GUI representation of it. The “knobs and buttons” are different, and it has its own look and feel. This is a completely new tool for the System i platform.

All examples in the tutorials are based on the sample QWQCENT database, which is shipped with 5733-QU2. This database contains four tables. Figure 3-8 on page 33 shows a Database map of the QWQCENT schema that was created using iSeries Navigator. The icons between the tables represent a foreign key constraint.

CENTURY versus QWQCENT: The sample database was called CENTURY on the pre-GA version of the product. It is called QWQCENT in the GA version of the product. For this reason, we might refer to the CENTURY database in some places, for what is now the QWQCENT database.

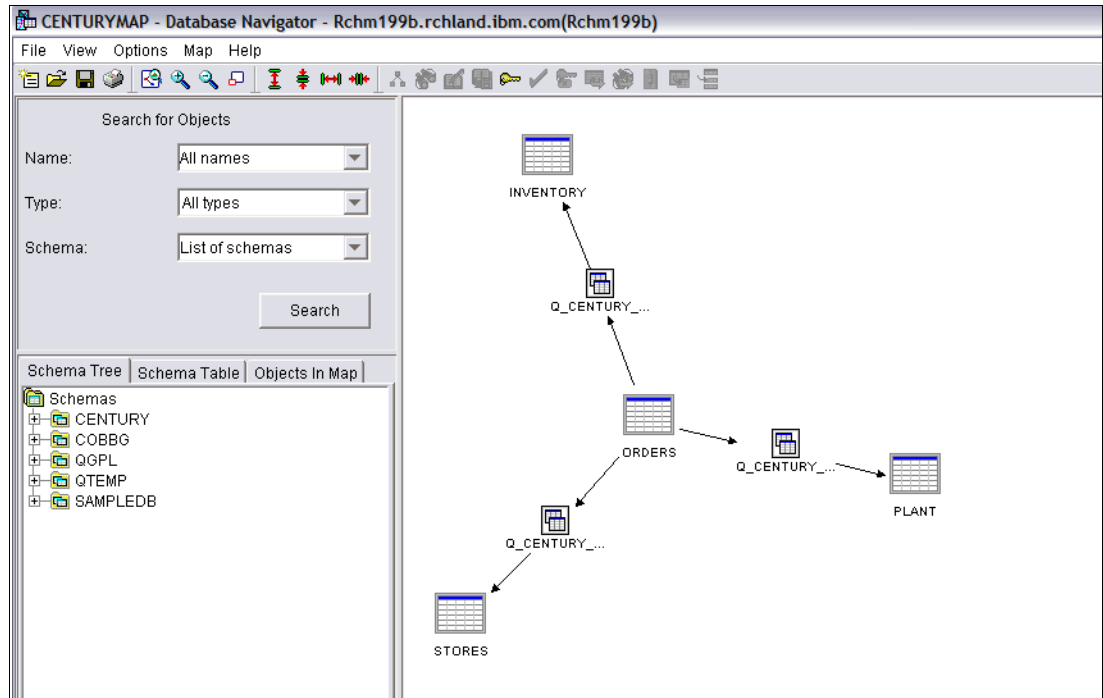


Figure 3-8 Database map of the QWQCENT (CENTURY) schema

For non-SQL programmers, the table shown in Figure 3-9 maps the equivalent terms between SQL and i5/OS.

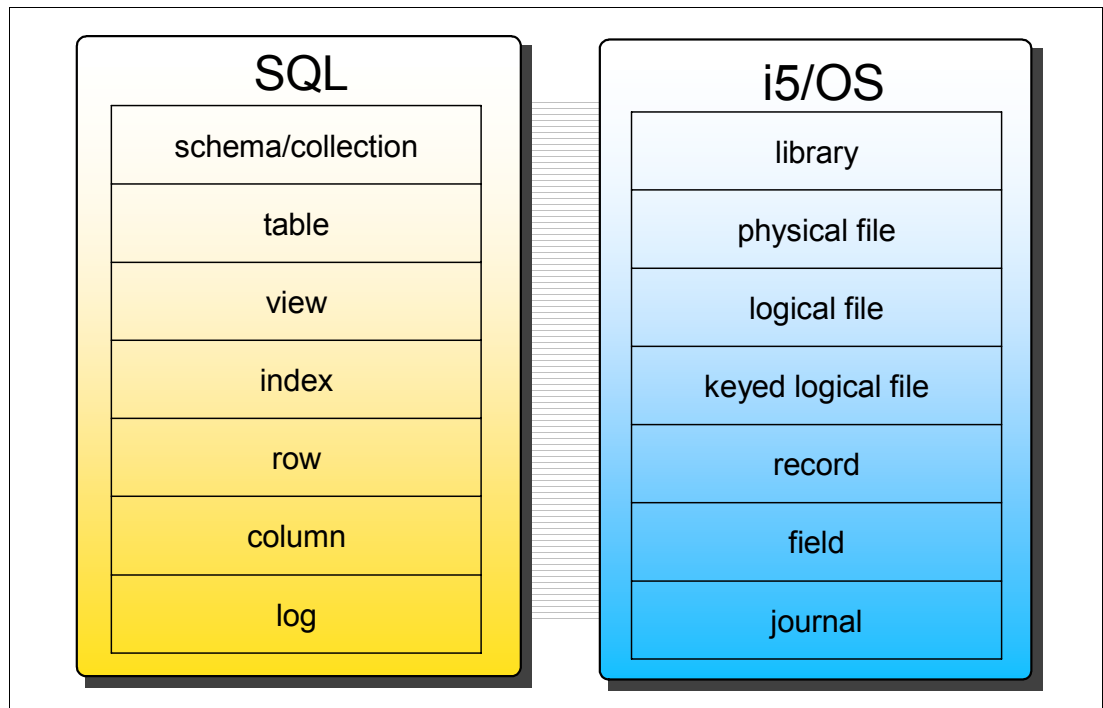


Figure 3-9 SQL and i5/OS terminology

3.4.1 Creating metadata

DB2 Web Query requires its own metadata to read from DB2 objects. This metadata is referred to as a *synonym*. First, the synonym must be created on the data that you want to query. Then it can be referenced in any number of reports. After you create the metadata once, you are not required to create it again for that same object unless the file layout changes. The metadata is a permanent object in the integrated file system. The metadata contains similar information to what DSPFFD contains. For more information about what is created on the system, see Appendix A, “Metadata in the integrated file system” on page 365.

To create metadata:

1. On the DB2 Web Query home page, expand the domain in which you want to work. Expand **Reports**. Right-click the specific report folder and select **Metadata** (Figure 3-10).

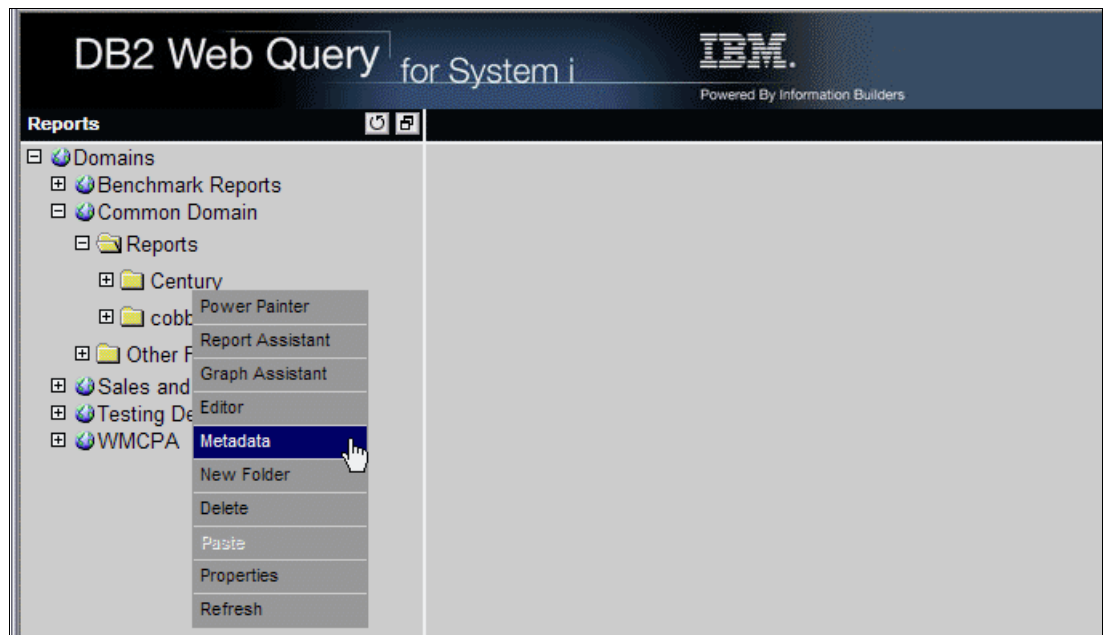


Figure 3-10 Selecting the Metadata option

2. In the Configuring Data Adapters pane (Figure 3-11), select the adapter that matches the data on which you will create metadata. In this example, we use DB2 cli, which is for a single member file. Click the **adapter type** and select **Create Synonym** as shown in Figure 3-11. If you have a connection to a remote database, select the database to which you want to connect from this panel. Notice that we click the ***LOCAL** connection.

To learn more about Data Adapters, refer to 1.5.5, “Data adapters” on page 13.

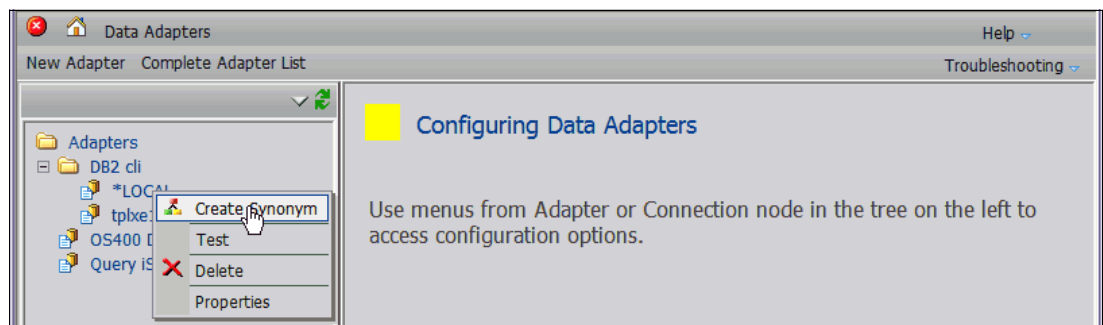


Figure 3-11 Creating a synonym

3. In the Select Synonym Candidates for DB2 cli pane (Figure 3-12), enter your collection name. Select the type of data that you want to query. In our example, we select **Tables** because we want to create metadata on a table in QWQCENT. In the Library field, type qwqcent. Click **Next**.

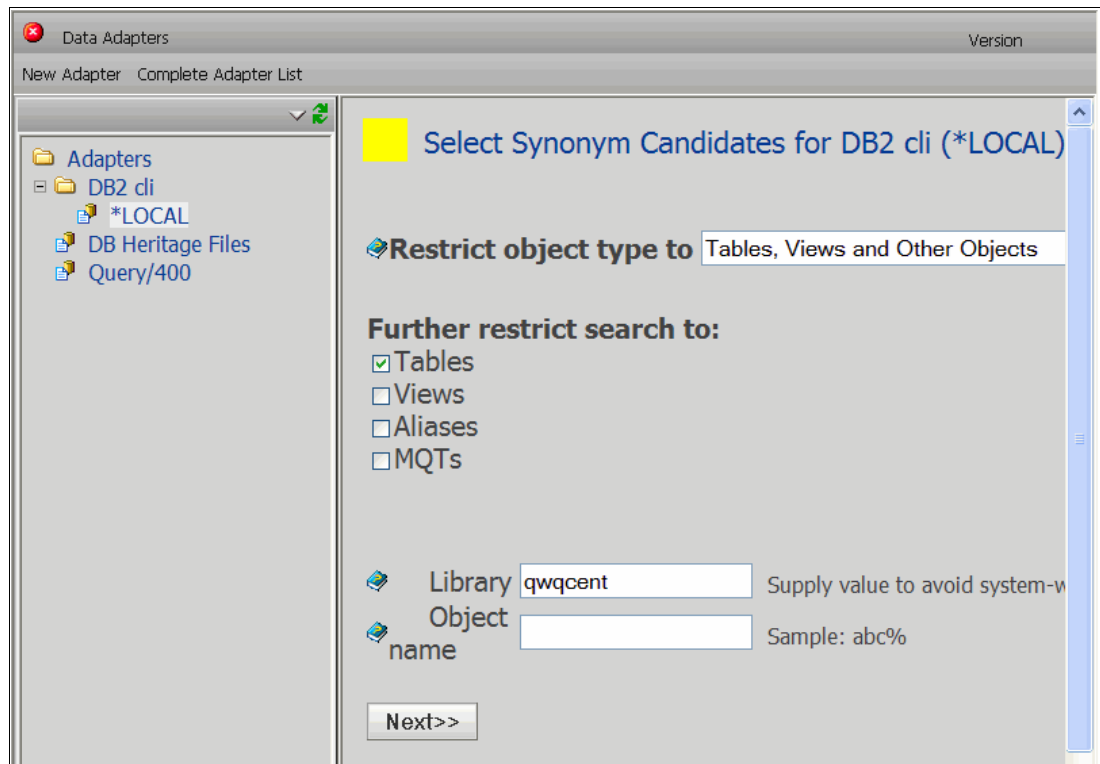


Figure 3-12 Selecting the schema or library

- The Create Synonym for DB2 cli pane (Figure 3-13) is displayed and shows all the different tables that reside on the QWQCENT (Century) schema. In this panel, select the table names for which you want to create reports.

You might want to include a prefix or suffix. In our example, cen_ is our prefix, because the table resides in library QWQCENT. We leave the suffix blank, but the developer has the choice to use it. Leave baseapp for Application. Click the **Create synonym** button.

Prefix and suffix (optional, yet recommended): The prefix and suffix are optional letters that you can add to your synonym name to provide extra meaning for you. Using a prefix or suffix is not required. It is up to the developer to decide whether it is necessary to use the prefix and the naming convention that is preferred. We recommend that you use the library name or an abbreviated version of the library name for the prefix. This way, when you create your reports, you can search on all metadata that starts with the library name.

Keep in mind that all metadata is displayed in one box called “Database Descriptions.” The box is rather small and does not list the metadata according to library.

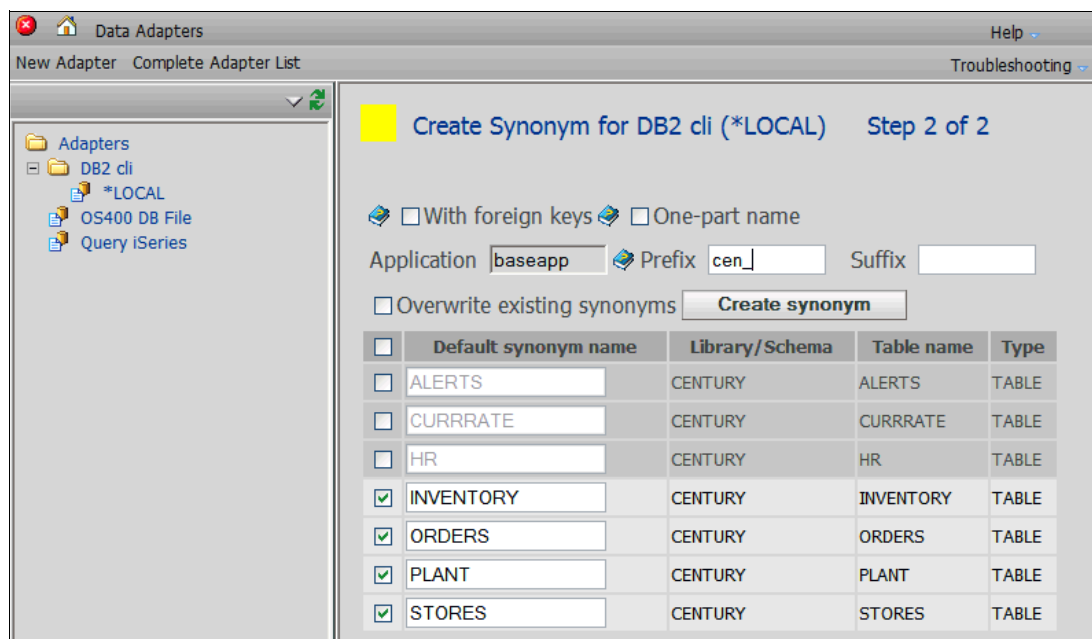


Figure 3-13 Selecting tables from a schema or library

The metadata takes a few seconds to create, depending on how many items you selected. After the processing is done, in the Status column, you see the message “Created successfully” as shown in Figure 3-14.

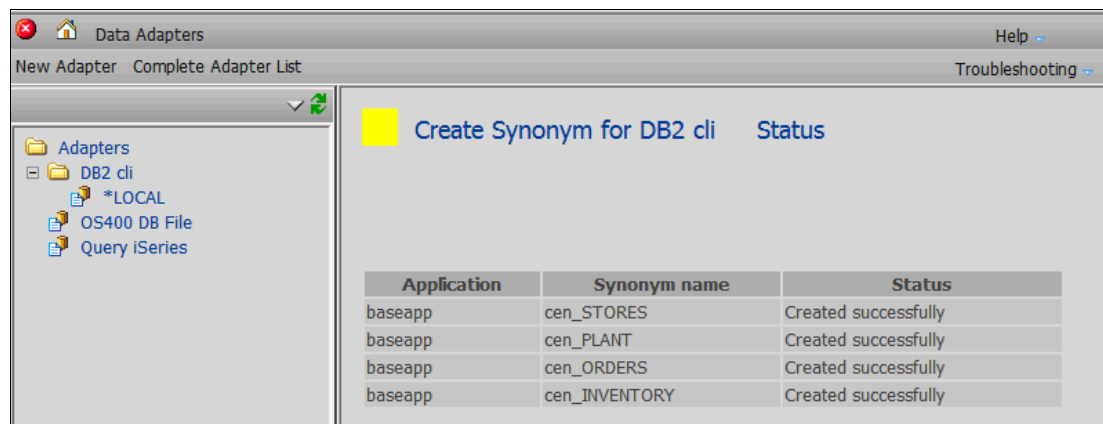


Figure 3-14 Created successfully message

The process of creating metadata is now complete. You are ready to begin the tutorials in Chapter 5, “Report Assistant” on page 63.

Miscellaneous considerations about creating metadata

The DB2 CLI adapter can create metadata on five different i5/OS object types: tables, views, aliases, stored procedures, and MQTs. Tables, views, aliases and MQTs are treated identically by DB2 Web Query. All of them can be used to filter, join, sort, define, compute, and access all the functionality of Report Assistant. After you create the metadata, you will be unable to tell which object type you are using in your report.

Creating metadata on a stored procedure

When you create metadata on a stored procedure, DB2 Web Query views the metadata as a result set. You can simply run the report and see the stored procedure result set returned. You can also use the functionality of Report Assistant to filter and manipulate the result set.

Multiple result sets: DB2 Web Query does support the return of multiple result sets from the stored procedure. In this case, it assigns a distinct segment name to the fields of each separate result set.

Considerations with multimember files

As noted in Table 1-2 on page 13, the DB2 CLI adapter sends SQL statements to i5/OS, where the DB2 Heritage File adapter sends the OPNQRYF CL commands. As indicated earlier, DB2 for i5/OS optimizes SQL and OPNQRYF differently. SQL can go down the codepath of the newer SQE, where OPNQRYF must still go down the CQE path. In general, SQE tends to be faster than CQE.

A limitation of SQL is that it can only read data from the first member in a file. Perhaps you have run into this limitation when writing SQL queries on multimember files before. One option is to consider using an SQL alias, which enables your query to use the DB2 CLI adapter and makes the query eligible for SQE. The alias is a permanent i5/OS object that “points” to a specific member in a file. It is easy to create, and after it exists, SQL treats the alias the same as it would a table. You simply substitute the alias name in any query where you would put a table name.

Here is an example of a CREATE ALIAS command in SQL:

```
CREATE ALIAS QGPL/MYALIAS FOR QGPL/MULTI_MBR_FILE (MBR_NUM_2)
```

You can run this command in the Run SQL Scripts window of iSeries Navigator or in STRSQL from the command line. STRSQL has a prompt for CREATE ALIAS that is intuitive.

You can create metadata on an alias as easily as you can on a regular table. Refer to 3.4.1, “Creating metadata” on page 34, which explains how to create DB2 Web Query metadata.

The DB2 Heritage File adapter also has the limitation of only being able to join on key fields. If you try to join on non-key fields using the multimember file adapter, you receive an error message (see Figure 3-15). This message might provide extra incentive to create an alias and use the DB2 CLI adapter.

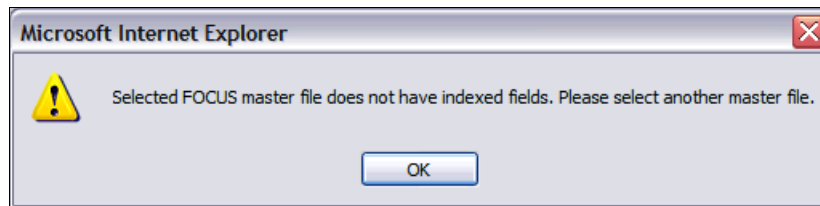


Figure 3-15 Error message regarding joining on non-key fields

What happens to the metadata if the i5/OS object changes

If you make changes to the format of the underlying i5/OS object, such as adding a field to a file, you are unable to use these changes in a DB2 Web Query report until the metadata is recreated. If you do not want to use the change in a DB2 Web Query report, the old metadata still works for the current report.

What happens to the metadata if the i5/OS object is deleted

If you delete the underlying table or the QRYDFN object, the DB2 Web Query report cannot run based on the metadata alone. If you attempt to run the report in DB2 Web Query, you receive an error message. You have the option to recreate the table or QRYDFN exactly as it was before. Therefore, you do not need to recreate the metadata. Or if you create a new, yet slightly different object, you must create new metadata on that objects.

Benefits of referential integrity for metadata

In this section, we explain the benefits, to DB2 Web Query, of having referential integrity defined in the database. We begin by briefly explaining referential integrity.

Referential integrity

A feature of a relational database is that it must support the definition and enforcement of referential integrity. If you do not fully grasp the meaning of referential integrity, you are not alone.

A database can consist of one large fact table and several smaller dimension tables. The dimension tables each contain a primary key that is referenced by a foreign key in the fact table. For our example, the primary key on the Inventory table is PROD_NUM, and the corresponding foreign key on the Order table is PROD_NUM. An entry in the fact table is not allowed unless there is a matching key in the dimension tables.

That is, you are not allowed to place an order for a product that is not confirmed to be in your inventory. The referential constraint does not permit an entry into the Orders table unless the entry has a value for PROD-NUM in the Inventory table. This prohibits unmatched, or “orphan”, entries in the Orders table.

Tip: You can also think of the small dimension tables as “parent” tables, and the large fact table as a “child” table. Do not let the part about the child table being larger confuse you. Think of it as a parent that has many children. The constraint does not let the child be born unless the parent already exists.

Benefit of using referential integrity in DB2 Web Query

When DB2 Web Query creates metadata, it recognizes files that are related through referential integrity and brings them all in together in one piece of metadata. When you write your report, you select a single synonym or table name that includes several files that are all related to each other. You do not need to manually define all the table joins. In order to have this convenience, you must have defined constraints on your tables.

If you do not currently have referential integrity in your database, we strongly recommend that you consider implementing it. Not only does referential integrity aid in more effective report writing in DB2 Web Query, but it also moves your business logic closer to the database level to reduce the programming effort. You do not need to understand SQL to add constraints to your database; you can do this easily by using iSeries Navigator. Referential integrity constraints can be added to files that are created with DDS and CRTPF, or they can be added to tables that are created with the CREATE TABLE statement.

Tip: If you want to add constraints by using the command line, use the ADDPFCST command. If you want to add constraints by using SQL, use the CREATE TABLE or ALTER TABLE statement.

The following brief example demonstrates how to set up referential integrity on our sample QWQCENT database. The Orders table is the fact table. The Inventory, Plant, and Stores tables are the dimensional tables. For our example, we set up referential integrity between the Orders table and the Inventory table. An item must exist in the Inventory table before it can have a “child” in the Orders table.

First we must create a primary key in INVENTORY, and then we can create the foreign key in ORDERS. The relationship between the two is called a *referential constraint*. The referential constraint starts at the child table and ensures that no child entry is inserted into the table unless an entry exists in the parent table.

Creating a primary key on INVENTORY field PROD_NUM

Open iSeries Navigator and follow these steps:

1. Expand the database container.
2. If the Century (QWQCENT) database is not currently displayed, right-click **Schemas** and then **Select Database to display**.

3. In the Select Schemas to Display window (Figure 3-16), in the Enter schema names field, type **Century** and click **Add**. Then click **OK**.

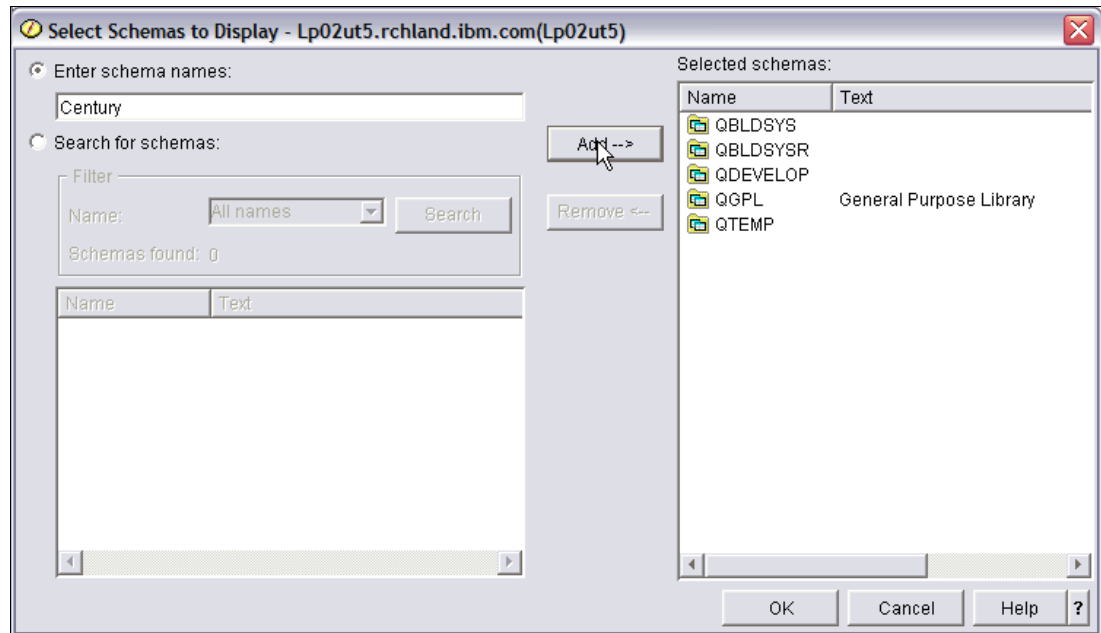


Figure 3-16 Select Schemas to Display window

4. In the left navigation bar of iSeries Navigator, expand the **CENTURY** database and click **Tables**. In the right pane, right-click the **INVENTORY** table and select **Definition** as shown in Figure 3-17.

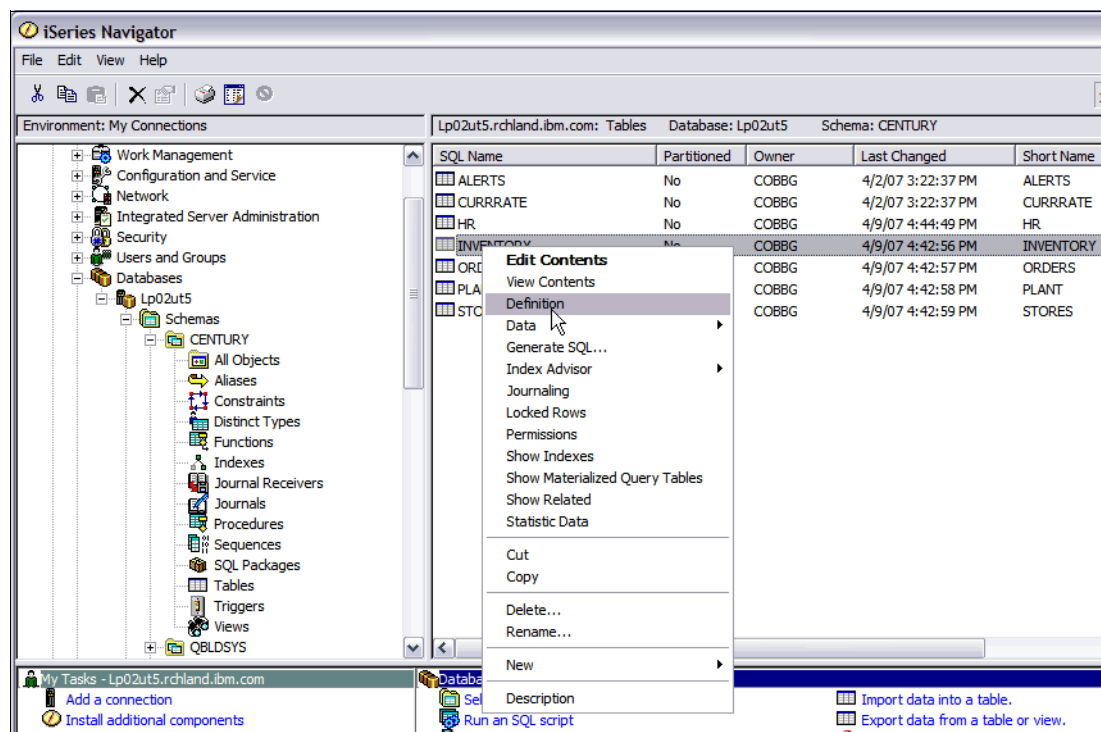


Figure 3-17 Selecting the Definition option

5. In the next window, from the tabbed selections, click the **Key Constraints** tab and then click **Add**.
6. In the New Key Constraint window (Figure 3-18), complete these steps:
 - a. Under Constraint type, select the **Primary Key** radio button.
 - b. Under Available columns, select the **ProductNumber** field and click the **Add** arrow button. Now the ProductNumber field is displayed under Selected columns.
 - c. Click **OK**.

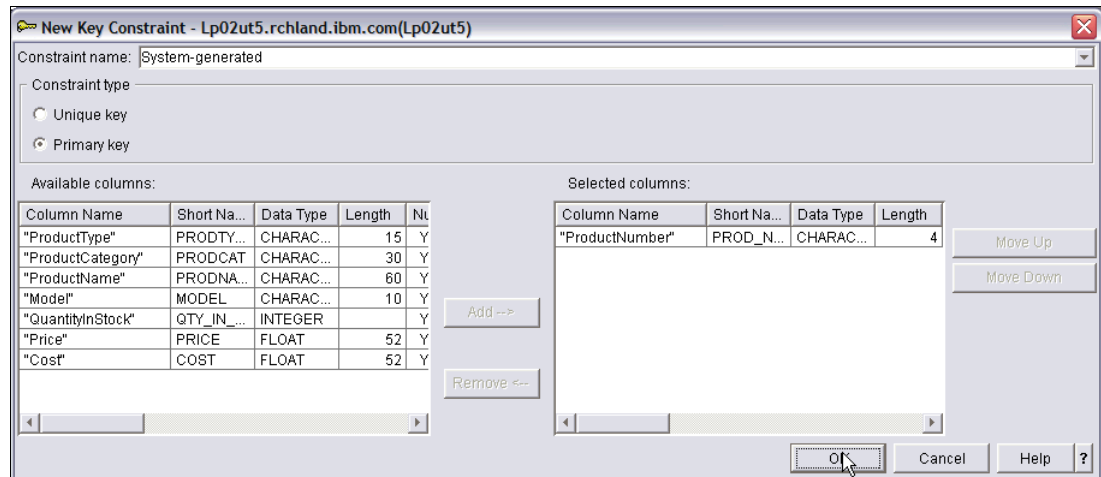


Figure 3-18 Key Constraints window

7. In the window that contains the various tabs (Figure 3-19), click **OK**.

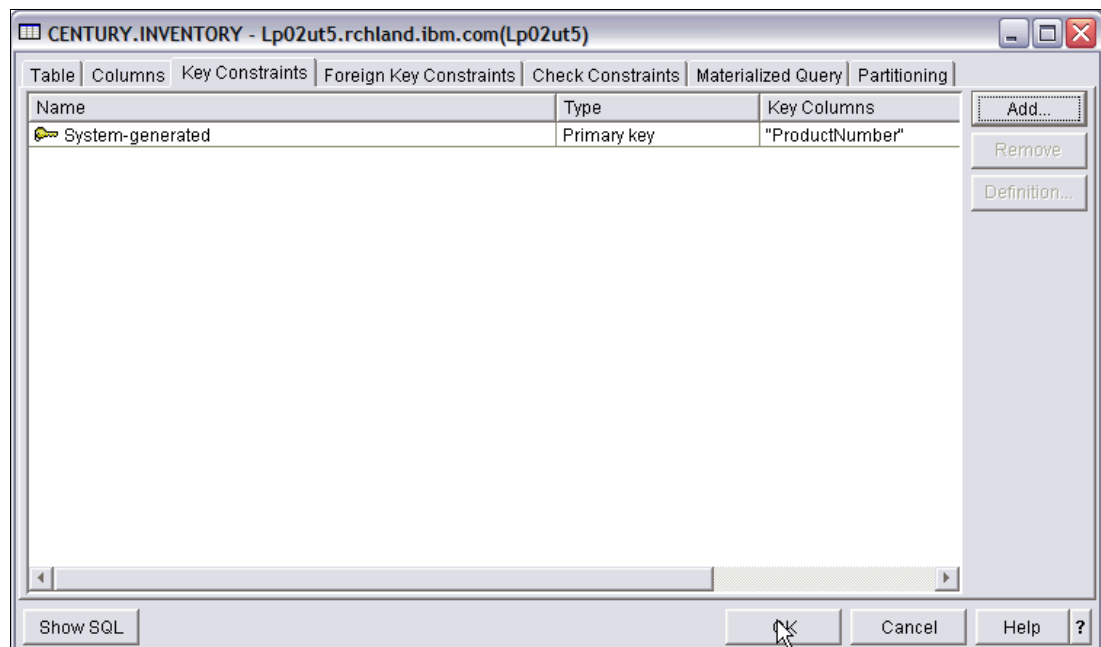


Figure 3-19 Key Constraints tab showing the ProductNumber field

You are now finished creating the primary constraint on the Inventory table. Next, we go to the Orders table and create a referential constraint. Remember that the Orders table checks the Inventory table to ensure that any new record also has a matching key value in Inventory.

1. Expand the **Century** schema and click **Tables**. Right-click the **Orders** table, and select **Definition**.
2. In the next window, click the **Foreign Key Constraints** tab. Click **Add**.
3. In the New Foreign Key Constraint window (Figure 3-20), complete these steps:
 - a. For Table name, select **Inventory**.
 - b. Under Available Columns, select **ProductNumber** and click **Add**. We know this is the primary key on the Inventory table because it is indicated under Key columns near the top of the window on the right.
 - c. Click **OK**.

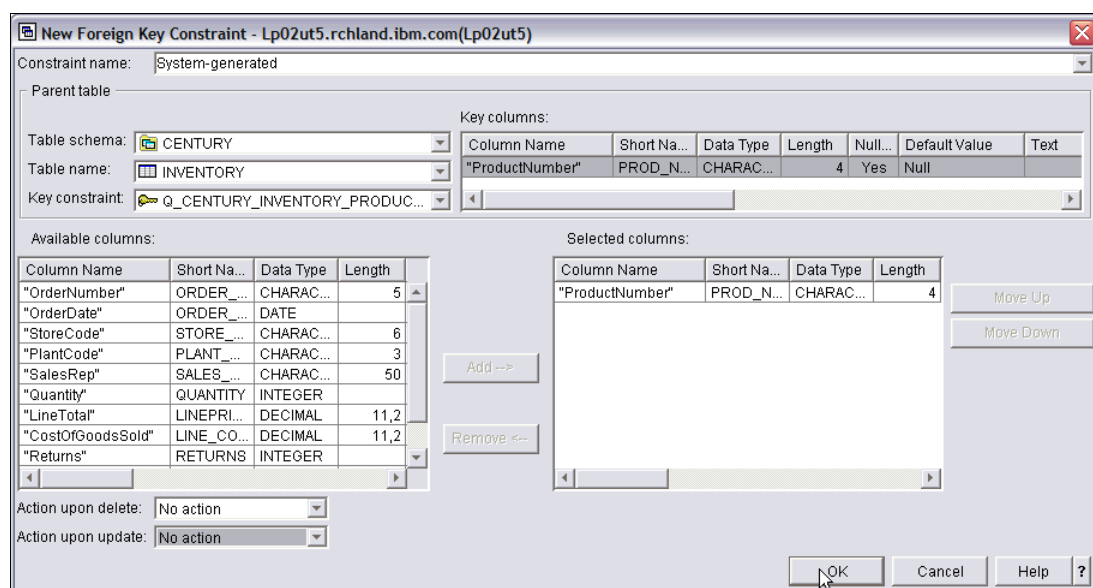


Figure 3-20 New Foreign Key Constraint window

You are now finished creating the referential constraint between the Inventory and Orders tables. You can now repeat this process with the two other dimension tables, Stores and Plant. The key fields on those tables are, respectively, Store_Code and Plant_Code. The Orders table has the corresponding foreign key for each table as it did for the Inventory table.

After you are done creating the other two referential constraints, your database is now ready for metadata creation.

Creating metadata on tables with referential integrity

The process to import a “cluster” of related tables into one piece of metadata is similar to creating a single piece of metadata. Review 3.4.1, “Creating metadata” on page 34. In the Create Synonym pane (Figure 3-21), notice the With foreign keys check box at the top of the page. This is the difference in the process. You must select **With foreign keys** so that the related tables are included in the same piece of metadata.

Then, the only table that you need to select is the fact table, which in our case is the **ORDERS** table. The dimension tables are automatically included. Then click **Create synonym**. You can check all the related tables if you want to, but the final metadata creation is the same.

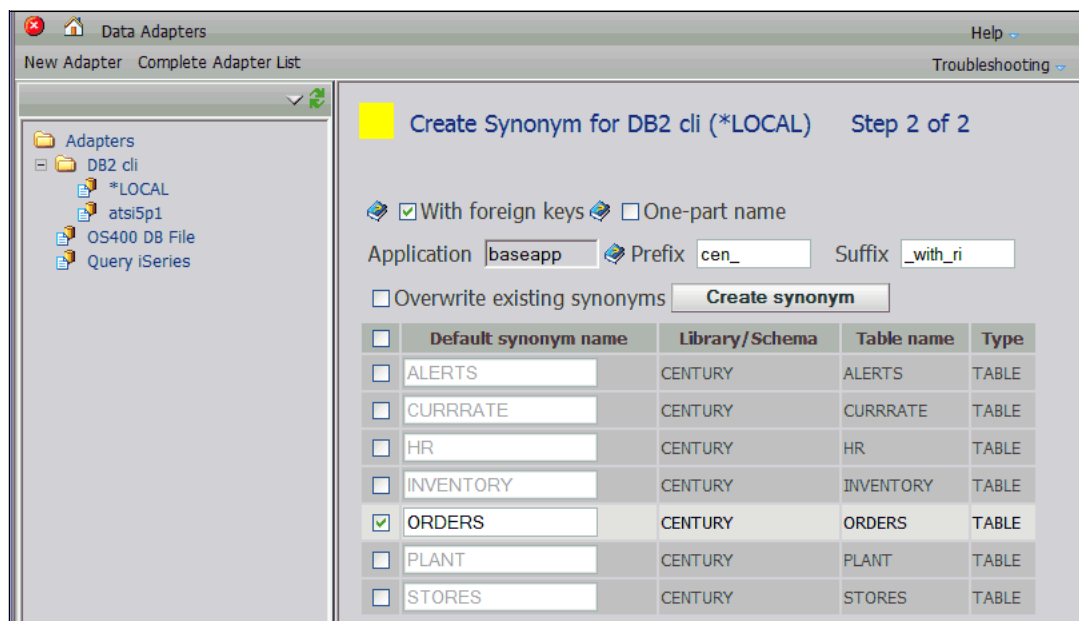


Figure 3-21 Creating metadata with referential integrity

Next the status pane (Figure 3-22) is displayed on which you see a status message of “Created successfully”. Notice that there is only one piece of metadata. You can close this panel.

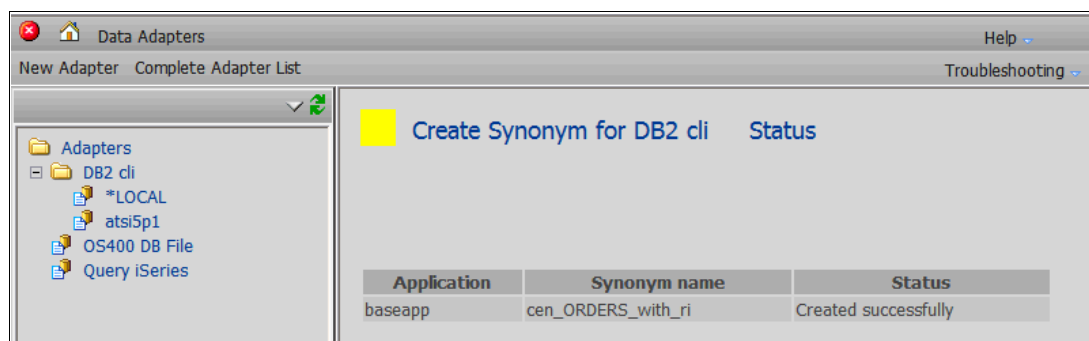


Figure 3-22 Created successfully message

When you first start to write the report based on the file cluster, you are presented with the metadata window. The related files are all under one name; the description says “Cluster xxx for table *yourFactTable*” like the example in Figure 3-23. The word “Cluster” in your metadata indicates that referential integrity has been used to create that metadata. All the related dimension tables with referential constraints defined appear in Report Assistant.

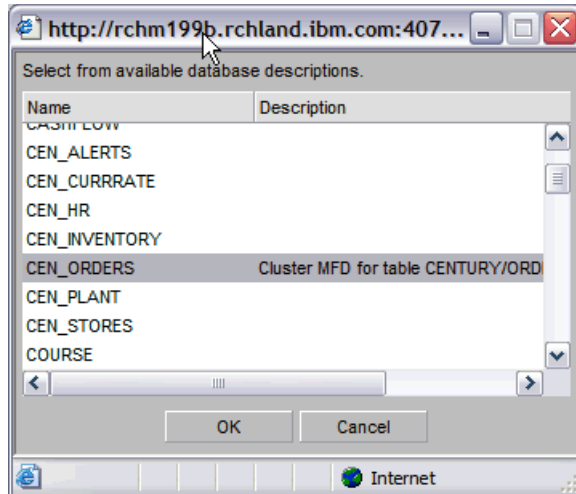


Figure 3-23 Cluster - Metadata with referential integrity

When you create a report, the list of columns can be presented alphabetically or by segment. In our case, segment means table. See Figure 3-24.

Field selection				
Report headings				
Selection criteria				
Join options				
Report options				
Available fields:				
Name	Alias	Format	Segment ^	
WARRANTYEXP	WarrantyExp	P13.2	T1_ORDERS	
STORECODE	StoreCode	A6	T1_ORDERS	
SHIPPINGCOST	ShippingCost	P13.2	T1_ORDERS	
SALESREP	SalesRep	A50	T1_ORDERS	
RETURNS	Returns	I11	T1_ORDERS	
QUANTITY	Quantity	I11	T1_ORDERS	
PRODUCTNUMBER	ProductNumber	A4	T1_ORDERS	
PLANTCODE	PlantCode	A3	T1_ORDERS	
ORDERNUMBER	OrderNumber	A5	T1_ORDERS	
ORDERDATE	OrderDate	YYMD	T1_ORDERS	
LINETOTAL	LineTotal	P13.2	T1_ORDERS	
COSTOFGOODSSOLD	CostOfGoodsSold	P13.2	T1_ORDERS	
QUANTITYINSTOCK	QuantityInStock	I11	T2_INVENTORY	
PRODUCTTYPE	ProductType	A15	T2_INVENTORY	
PRODUCTNUMBER	ProductNumber	A4	T2_INVENTORY	
PRODUCTNAME	ProductName	A60	T2_INVENTORY	
PRODUCTCATEGORY	ProductCategory	A30	T2_INVENTORY	
PRICE	Price	D20.2	T2_INVENTORY	
MODEL	Model	A10	T2_INVENTORY	
COST	Cost	D20.2	T2_INVENTORY	
PLANTTELEPHONE	PlantTelephone	A12	T3_PLANT	
PLANTTELEFAX	PlantTelefax	A12	T3_PLANT	
PLANTSTATE	PlantState	A25	T3_PLANT	
PLANTREGION	PlantRegion	A25	T3_PLANT	
PLANTPOSTALCODE	PlantPostalCode	A10	T3_PLANT	
PLANTNAME	PlantName	A15	T3_PLANT	
PLANTEMAIL	PlantMail	A40	T3_PLANT	
PLANTCOUNTRY	PlantCountry	A15	T3_PLANT	

Figure 3-24 Field selection sorted by table

3.4.2 Setting up a cross-system join

The base version of DB2 Web Query provides the ability to access tables and views on remote systems. This means that you can create reports against data that resides on the following systems:

- ▶ The local system
- ▶ A remote System i machine or logical partition (LPAR)
- ▶ A combination of both the local system and a remote System i machine or LPAR

This feature provides the foundation for creating a report with a federated or cross-system join. A *cross-system join* is a join in which the underlying base tables reside on multiple System i machines. The restriction here is that all databases (both local and remote) must be DB2 for i5/OS. If you want to access data on another platform such as DB2 for Linux®, UNIX®, or Windows (LUW), you must acquire the specific add-on data adapter for that database.

In the following task, we explain how to create a connection to a remote DB2 on a System i machine. If you already defined a remote database using the WRKRDBDIRE command, you can skip the first three steps.

1. Open iSeries Navigator.
2. In iSeries Navigator, expand the **desired connection name**. Under the expanded connection, right-click **Databases** and select **New** → **Relational Database Directory Entry** as shown in Figure 3-25.

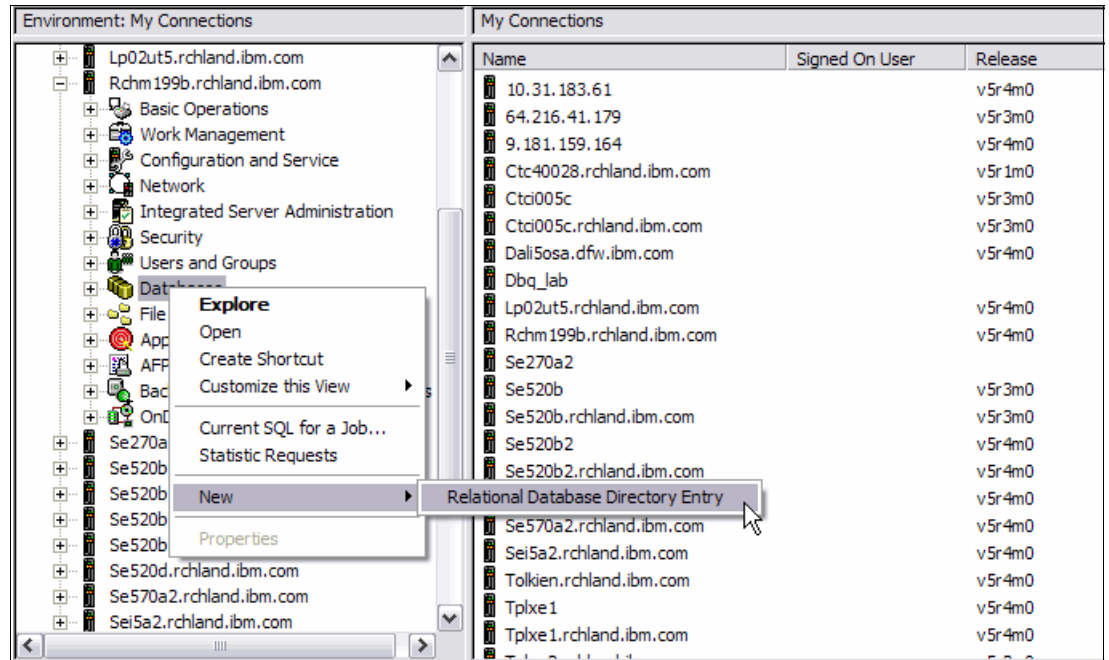


Figure 3-25 Selecting the option to create a new Relational Database Directory Entry

3. In the Add RDB Directory Entry (ADDRDBDIRE) window (Figure 3-26), specify the following options:
 - Relational database

This is the logical name to assign to the relational database. It is usually the name of the remote system or partition.
 - Remote location name or address

This is either the fully qualified system or partition name or the IP address.
 - Remote location type

This refers to the protocol that is being used to communicate between the two systems or partitions. In most cases, you specify *IP for this setting.

Click **OK**.

The screenshot shows the 'Add RDB Directory Entry (ADDRDBDIRE)' dialog box. It has a menu bar with 'File', 'Edit', 'View', and 'Help'. The main area contains the following fields and options:

- Entry:**
 - Relational database: Character value
 - Relational database alias: Character value
- Remote location:**
 - Name or address: Character value
 - Type: ☐ *SNA ☒ *IP
- Text:** Character value
- Port number or service program:** Character value
- Remote authentication method:**
 - Preferred method:
 - Allow lower authentication: ☒ *ALWLOWER ☐ *NOALWLOWER
- Device:**
 - APPC device description: Name
 - Local location: Name
 - Remote network identifier: Name
 - Mode: Name
 - Transaction program: Character value
- Application requester driver:**
 - Program:
 - Library: Name

At the bottom right, there are buttons for 'OK', 'Cancel', 'Help', and a question mark icon.

Figure 3-26 Add RDB Directory Entry window in iSeries Navigator

Tip: Another option is to use the CL command ADDRDBDIRE.

4. Open your Web browser to the DB2 Web Query console on port 11331, for example:
<http://yoursystemname:11331/webquery>
5. Enter your System i user ID and password.
6. Navigate down to any folder under Reports, right-click, and select the **Metadata** option.
7. Select **Data Adapter** from the menu bar.
8. In the Data Adapters window (Figure 3-27), click **DB2 cli** and select **Add connection**.

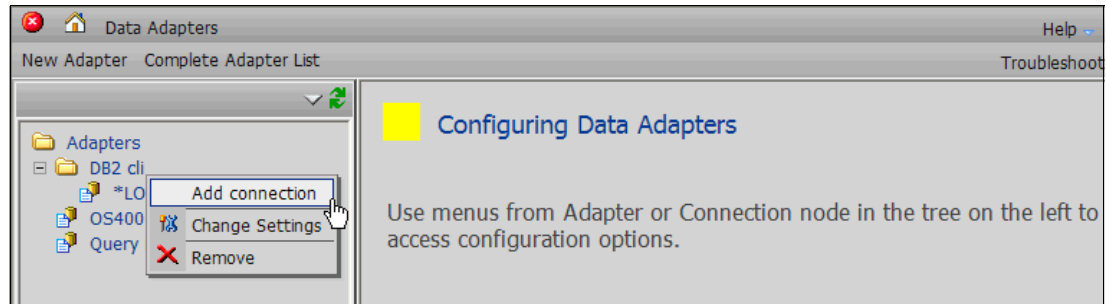


Figure 3-27 Adding a connection to data adapters

9. In the Add Connection for DB2 cli pane (Figure 3-28 on page 49), specify the following options:
 - Connection Name
 This is the logical name that is used to identify this particular set of connection attributes. It is usually the remote system or partition name, but it does not have to match any value that is specified during the Add RDB Directory Entry step (step 3 on page 47).
 - Datasource
 This is the DB2 database name that is used for this connection. The name must match the Relational Database value that is specified during the Add RDB Directory Entry step.
 - Security
 There are three methods by which a user can be authenticated when connecting to a DB2 database server:
 - Explicit
 The user ID and password are explicitly specified for each connection and passed to DB2, at connection time, for authentication.
 - Password Passthru
 The user ID and password received from the client application are passed to DB2, at connection time, for authentication.
 - Trusted
 The adapter connects to DB2 as a Windows login using the credentials of the Windows user, who impersonated by the server data access agent.
 - User
 For Explicit security only, this field is for the i5/OS user profile.

- Password

For Explicit security, this field is for the password that is associated with the specified i5/OS user profile.

Click the **Configure** button.

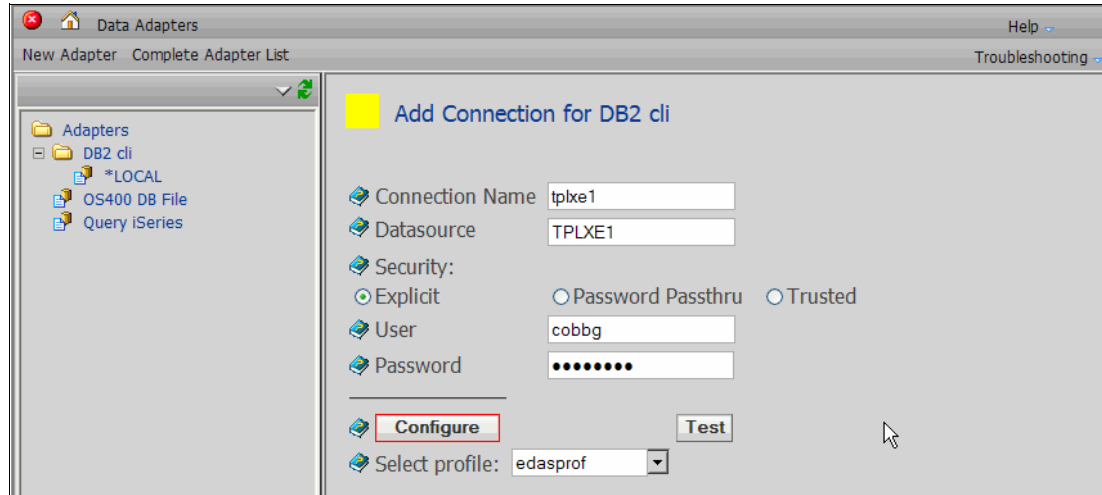


Figure 3-28 Add Connection for DB2 cli pane

10. Create the synonyms for the remote tables or views that you want include in your reports:

- In the Data Adapters window (Figure 3-29), click the remote connection and select **Create Synonym**.

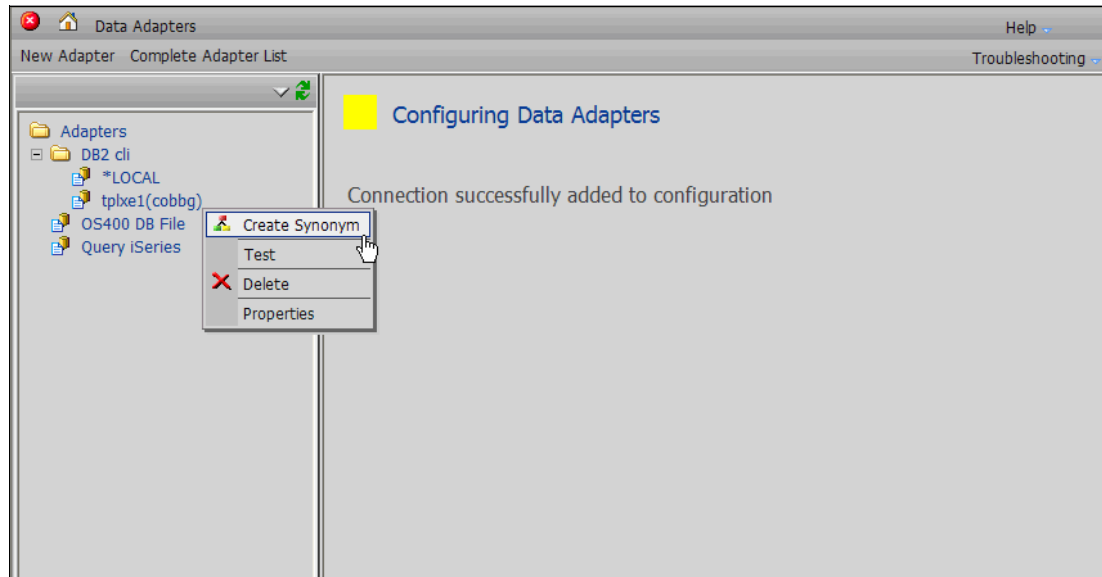


Figure 3-29 Selecting the Create Synonym option

- b. In the Select Synonym Candidates for DB2 cli pane (Figure 3-30), for Restrict object type to, select **Tables** and **Views**. Then specify the name of the Library (schema) that contains the database objects that you want to include in the cross-system join. Click **Next**.

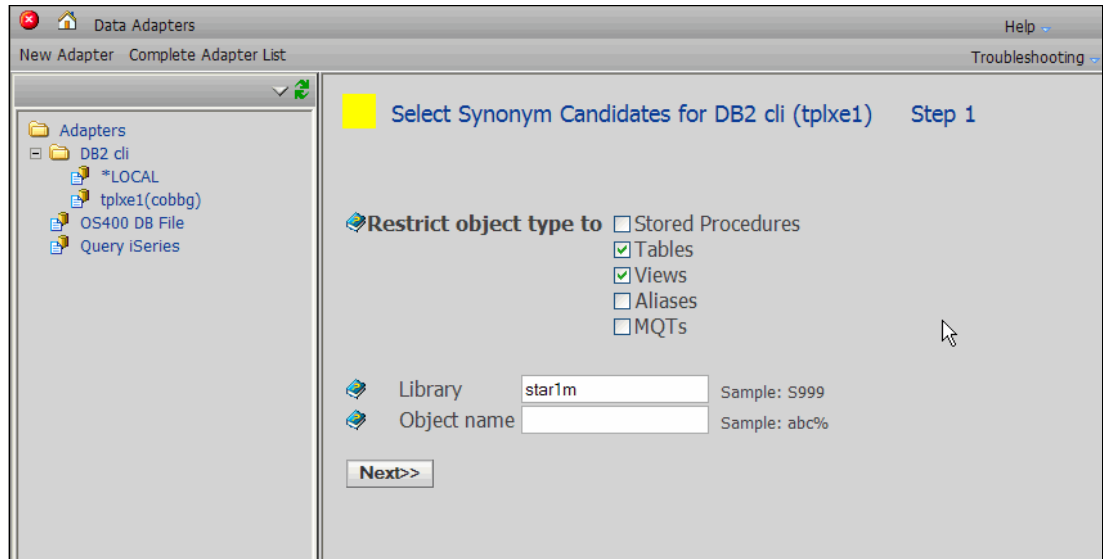


Figure 3-30 Select Synonym Candidates for DB2 cli pane

- c. In the Create Synonym for DB2 cli pane (Figure 3-31), specify the prefix (recommended format is connectionName_libraryName_) and select the tables, views, or both that you want to include in the cross-system join. Click **Create synonym**.

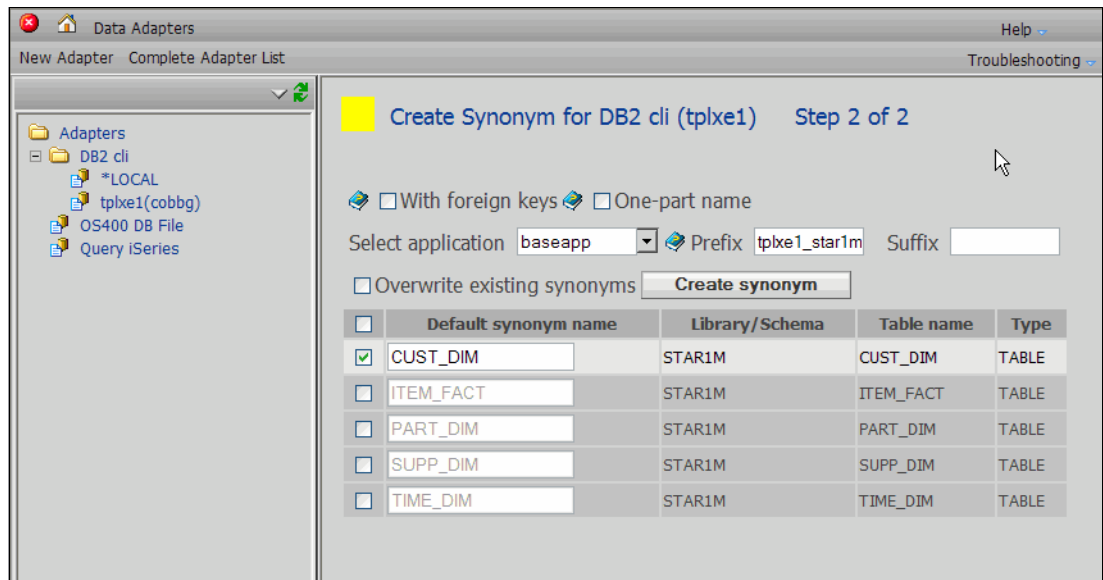


Figure 3-31 Create Synonym for DB2 cli pane

11. Include the table in your reports by using a cross-system join:
 - a. In the DB2 Web Query home page, under the desired folder, select **Report Assistant**.
 - b. In the Select from available database descriptions panel, find and select the synonym for the *local* table; do not select remote synonym just yet.
 - c. In the Report Assistant window (Figure 3-32), click the **Join options** tab.
 - d. On the Join options tab, complete these steps:
 - i. Click **New**.
 - ii. In the Web Page Dialog that opens, find and select the synonym for the *remote* table and click **OK**. Notice that the Join options page changes.

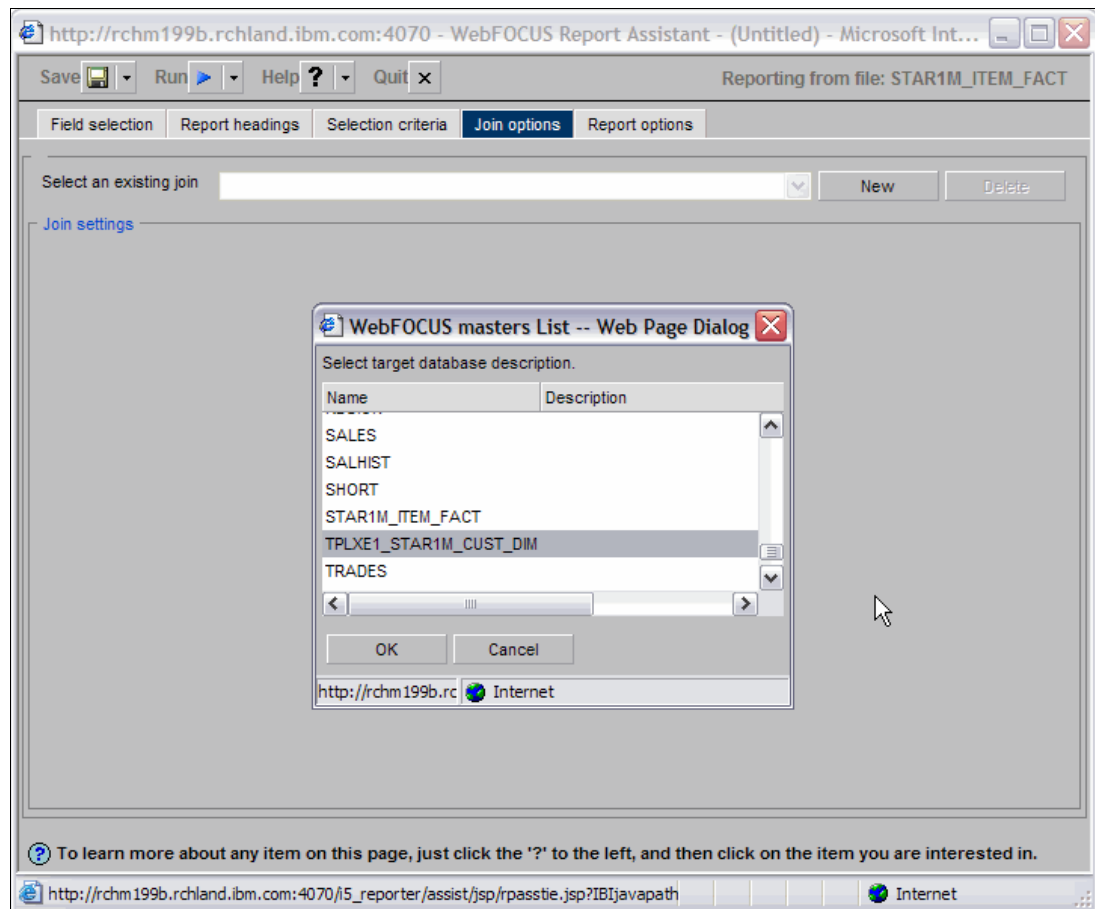


Figure 3-32 Selecting the target database description

- iii. Under Join Settings, as shown in Figure 3-33, specify the Join Type, Description, Instances, and Join fields options.

Note: For more information about joining tables, see 5.5, “Joining tables (reference only)” on page 106.

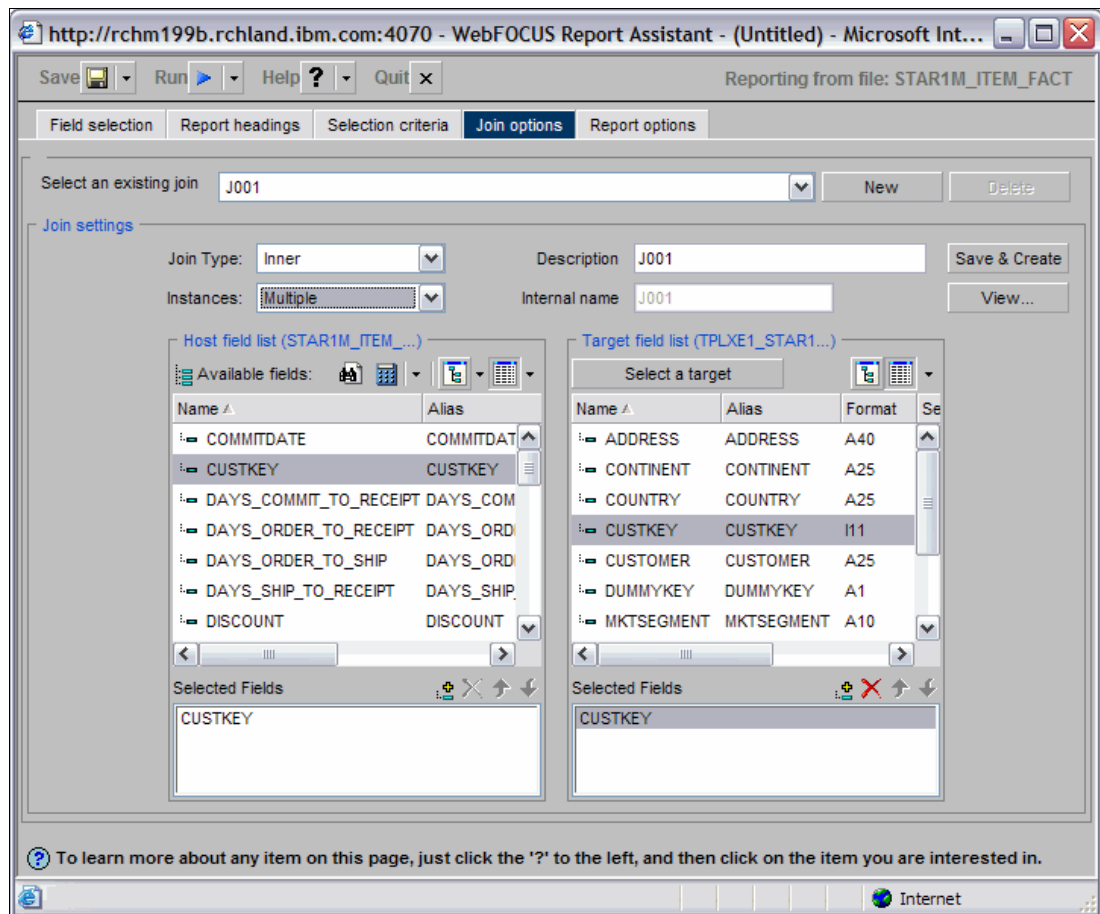


Figure 3-33 Specifying the join options

12. Click the **Field Selection** tab. At this point, you can finish the report by specifying the desired options such as Sort by and Sum as shown in Figure 3-34. All of the fields from the joined files are displayed in the list of fields to select from. Notice that the fields are color coded to clearly identify the files from which they originated. See Chapter 5, “Report Assistant” on page 63, for more details about report writing.

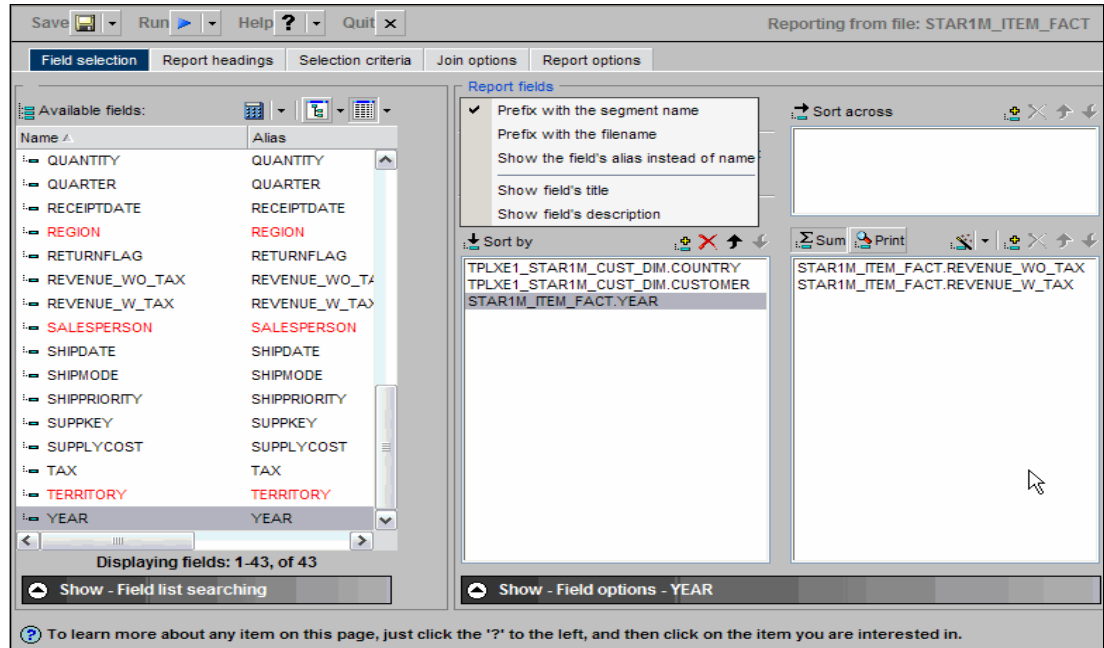


Figure 3-34 Field selection tab

3.5 Basic user

The basic user employs DB2 Web Query to run reports, so in this section, we explain how to run a report. Before you continue, ensure that you have read 3.2, “Logging in” on page 26, and are logged on to the DB2 Web Query home page. You only see the domains to which you are authorized. If you do not see a domain that you need, contact your DB2 Web Query administrator.

To run a report, expand the domain. Then expand the **Reports** folder and the subfolder. The subfolders inside of this report categorize the reports within the domain. After you expand the subfolder, double-click the report that you want to run. The Processing Request panel is displayed while your results are retrieved from the server. Then the results are displayed in the right panel (see Figure 3-35).

The screenshot shows the DB2 Web Query for System i interface. On the left is a tree view under 'Reports' with the following structure:

- Domains
 - Common Domain
 - Reports
 - Other Files
 - Testing and Development
 - Reports
 - Reports
 - Active Report
 - Chapter 4 detailed

The main content area displays 'PAGE 1' and a report titled 'Gross Profit by Product Category'. The report is a table with the following data:

Product Type	Gross Profit Rank	Product Category	Line Total	Cost of Goods
Audio	1	Speakers	84717053.00	2468
	2	Audio Systems	122345680.00	8228

Figure 3-35 Running a report

Keep in mind that the results of your report are returned to your PC and stored in your browser. This affects the time that it takes for your results to appear. If your report returns several thousands of records, you will notice more delay than if your report returns ten records. If you think the report returns more records than you can use, ask your developer to limit the number of records that are returned to you. We explain how to do this in 5.2.3, “Sum and detail reports” on page 68.

Tutorials for DB2 Web Query

This part and Part 3, “Tutorials for DB2 Web Query optional features” on page 173, are structured as basic tutorials. These examples are not intended to replace reference manuals or customer education classes. The intent is to provide hands-on exercises that help you get started and demonstrate the more common reporting techniques.

This part includes the following chapters:

- ▶ Chapter 4, “Getting started with the tutorials” on page 57
- ▶ Chapter 5, “Report Assistant” on page 63
- ▶ Chapter 6, “Graph Assistant” on page 109
- ▶ Chapter 7, “Power Painter” on page 153

Chapter 4, “Getting started with the tutorials” on page 57, lays the groundwork for the following tutorials. In Chapter 4, “Getting started with the tutorials” on page 57, we explain the tables that we are going to use in future chapters and set up security. If this has already been done for you, you can go straight to Chapter 5, “Report Assistant” on page 63.

The tutorials all use tables in the QWQCENT library. This library can also be referenced as the “Century” library. Century is the fictitious electronics company that is referenced throughout these tutorials. Specifically we use the following tables:

- ▶ ORDERS
- ▶ INVENTORY
- ▶ PLANT
- ▶ STORES

ORDERS is the central table. It contains foreign keys or links to the other three tables. In our examples, you will see that we only directly refer to the ORDERS table. Thanks to the knowledge and understanding that DB2 Web Query has DB2 for i5/OS, the other three tables are automatically made available whenever ORDERS is referenced. See the following figure.

CENTURY.INVENTORY
System Name:INVENTORY

Column Name	...	Data ...	Length
"ProductNumber"	1	CHAR...	4
"ProductType"		CHAR...	15
"ProductCatego..."		CHAR...	30
"ProductName"		CHAR...	60
"Model"		CHAR...	10
"QuantityInStock"		INTEG...	
"Price"		DECIM...	11,2
"Cost"		DECIM...	11,2

CENTURY.STORES
System Name:STORES

Column Name	...	Data ...	Length
"StoreCode"	1	CHAR...	6
"StoreName"		CHAR...	30
"Country"		CHAR...	15
"Region"		CHAR...	25
"State"		CHAR...	25
"City"		CHAR...	25
"PostalCode"		CHAR...	10
"Address"		CHAR...	40
"Telephone"		CHAR...	12
"Telefax"		CHAR...	12
"eMail"		CHAR...	40

CENTURY.ORDERS
System Name:ORDERS

Column Name	...	Data ...	Length
"OrderNumber"		CHAR...	5
"ProductNumber"		CHAR...	4
"OrderDate"		DATE	
"StoreCode"		CHAR...	6
"PlantCode"		CHAR...	3
"SalesRep"		CHAR...	50
"Quantity"		INTEG...	
"LineTotal"		NUME...	20,2
"CostOfGoodsS..."		NUME...	20,2
"Returns"		INTEG...	
"WarrantyExp"		NUME...	20,2
"ShippingCost"		NUME...	20,2

CENTURY.PLANT
System Name:PLANT

Column Name	...	Data ...	Length
"PlantCode"	1	CHAR...	3
"PlantName"		CHAR...	15
"PlantCountry"		CHAR...	15
"PlantRegion"		CHAR...	25
"PlantState"		CHAR...	25
"PlantCity"		CHAR...	25
"PlantPostalCo..."		CHAR...	10
"PlantAddress"		CHAR...	40
"PlantTelephone"		CHAR...	12
"PlantTelefax"		CHAR...	12
"PlantEmail"		CHAR...	40

Map of tables in QWQCENT - Century library

In DB2 for i5/OS, the following foreign keys are defined in the ORDERS table:

- ▶ ProductNumber is the key to the INVENTORY table.
- ▶ StoreCode is the key to the STORES table.
- ▶ PlantCode is the key to the PLANT table.



Getting started with the tutorials

In this chapter, we set up the System i platform and the user profiles in preparation for the tutorials that follow. We explain many of the steps in this chapter in more detail in Chapter 3, “DB2 Web Query fundamentals” on page 25. They are included here to give you complete, step-by-step instructions and to enable the tutorials to stand alone. For more understanding behind items, such as the security, domains and Hrefs, refer to Chapter 3, “DB2 Web Query fundamentals” on page 25.

In preparation to start using the tutorials, you must authorize your user profile to create tables in the Common Domain. Then you must create a folder to store all your work. Finally you must define all the tables that we will query to DB2 Web Query through the Create Synonym process.

1. Before you start using the tutorials, register your user profile as a valid licensed DB2 Web Query user. Enter the following command:

```
CALL QWEBQRY76/WQADDLIC 'USERPROFILE'
```

USERPROFILE references your System i user profile name and must be specified in uppercase letters.

For more details about registering and removing users, see 3.1, “Registering named users” on page 26.

2. Incorporate the user profile to a group profile called MDUNTITLED. When you install DB2 Web Query, the system creates a group profile named MDUNTITLED. To create reports in the Common Domain, you must be a member of either MDUNTITLED. As a member of the MDUNTITLED group, you are authorized to create new reports in the default Common Domain. For more details about security and authorizations, see 3.3.3, “Assigning authority to users” on page 31.

MRADMIN group: Being a member of the MRADMIN group instead of MDUNTITLED also works for these tutorials. The MRADMIN group profile allows a user to create new reports in all domains and to create new domains. This additional authorization is not required for the tutorials.

3. Open a browser.

We tested these steps using both Internet Explorer and Firefox, but use Internet Explorer for the tutorials presented in this book.

4. Since DB2 Web Query uses port 11331, point your browser to the following URL (Figure 4-1):

```
http://systemi_name:11331/webquery
```

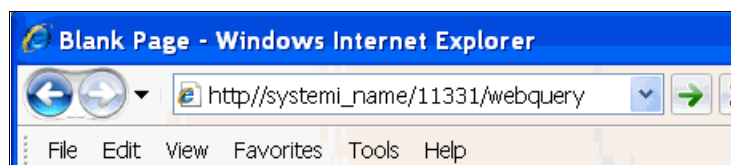


Figure 4-1 URL to initiate DB2 Web Query

5. Enter your System i user ID and password (Figure 4-2).

Figure 4-2 Initial logon window

6. In the main DB2 Web Query home page, right-click **Common Domain** and select **Properties** to see the domain properties.
7. The Domain Properties window (Figure 4-3) opens. The Common Domain is where we store our reports. One of the properties is the Href or the system identifier. For the Common Domain, this is always “untitled”. By previously including yourself in the group MDUNTITLED, you now have developer authority (MD) in the domain UNTITLED. Close this window.

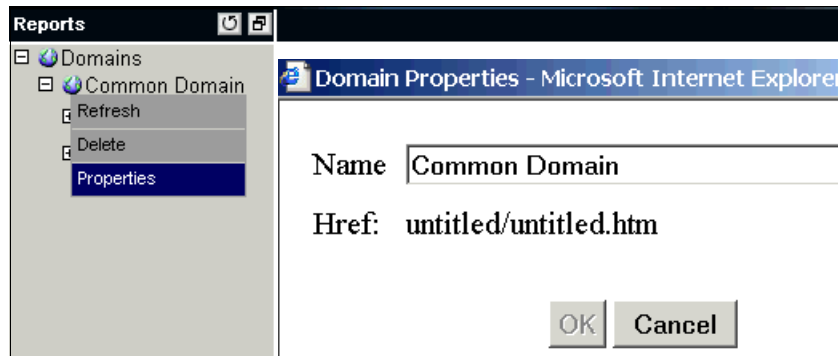


Figure 4-3 Common Domain properties

8. Expand **Common Domain**, right-click the **Reports** folder, and select **New Folder**.
9. In the New Folder window (Figure 4-4), create a new folder called **Tutorials**, which is where we store all of our work. Click **Save**.

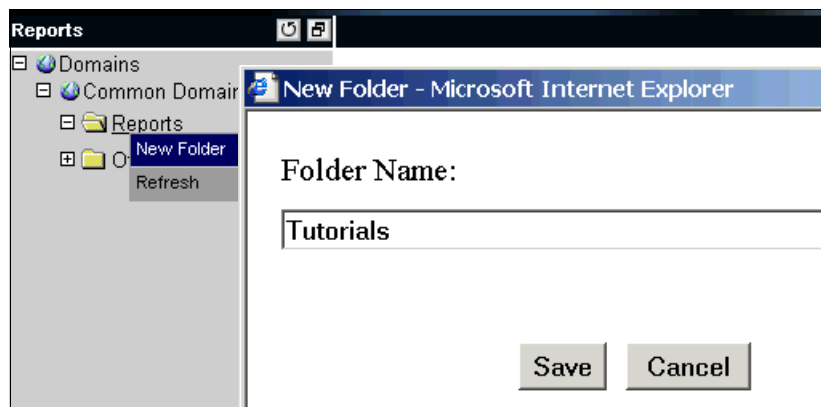


Figure 4-4 Creating a new folder

10. Before you can create a report on the System i machine, you must tell DB2 Web Query which tables the users can query. DB2 Web Query then creates the metadata or synonyms to describe the selected tables. For more details, see 3.4.1, “Creating metadata” on page 34.

In the left navigation area (Figure 4-5), right-click **Tutorials** and select **Metadata**.

Note: Although you select a folder to enable the submenu that contains metadata, the metadata is not restricted to that folder. After the metadata is created, it is available system wide and to all reports.

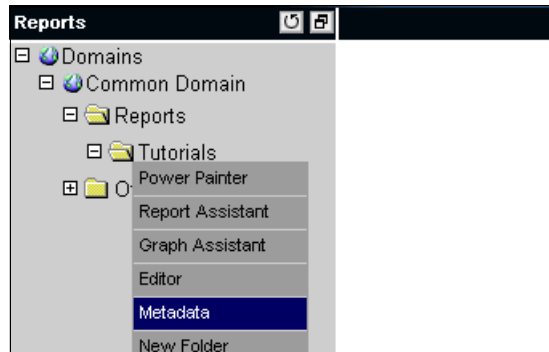


Figure 4-5 Creating metadata

11. In the Data Adapters window (Figure 4-6), right-click ***LOCAL**, which is the System i where the table resides, and select **Create Synonym**.

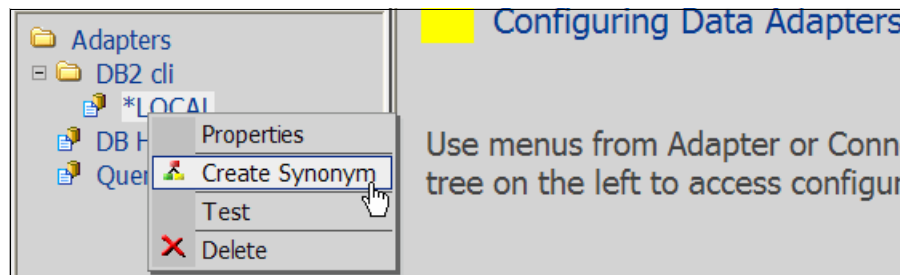


Figure 4-6 Creating a synonym

Tip: If you have other System i machines, logical partitions (LPARs), or independent auxiliary storage pools (IASPs) in your network, you can add a connection to these systems. Then the database names will be displayed under *LOCAL. Any of these remote DB2 tables are then available for use. These additional databases must be defined in the relational database directory table on the System i machine.

12. Enter the library where the tables reside. The library that we use throughout the tutorials is the QWQCENT library.

Note: There are three types of System i objects that you can query with the base DB2 Web Query product.

- ▶ *DB2 cli* refers to the majority of your tables created with either an SQL CREATE TABLE statement or by using data description specifications (DDS) and a CRTPF command.
- ▶ *DB2 Heritage files* refers to older tables that have multiple record types or multiple members. Tables described with this adapter use the OPNQRYF command on the System i machine. It then directs the query to the older, and typically slower, database optimizer (Classic Query Engine (CQE)).
- ▶ *Query/400* imports the existing Query/400 definitions and converts them to both table and report definitions.

13. In the Select Synonym Candidates on DB2 cli pane (Figure 4-7), complete these steps:

- a. For Restrict object type to, specify **Tables, Views, and Other Objects**.
- b. Under Further restrict search to:, select the **Tables** option. In this example, we clear the Views check box. This leaves us with a concise list that contains only our physical files or tables.

SQL views: If you created SQL views or logical files and want to query them, you must leave Views selected. If this library is an SQL collection or schema, selecting Views also displays all of the SQL catalog views in the library.

- c. For Library, type qwqcent.
- d. Click **Next**.

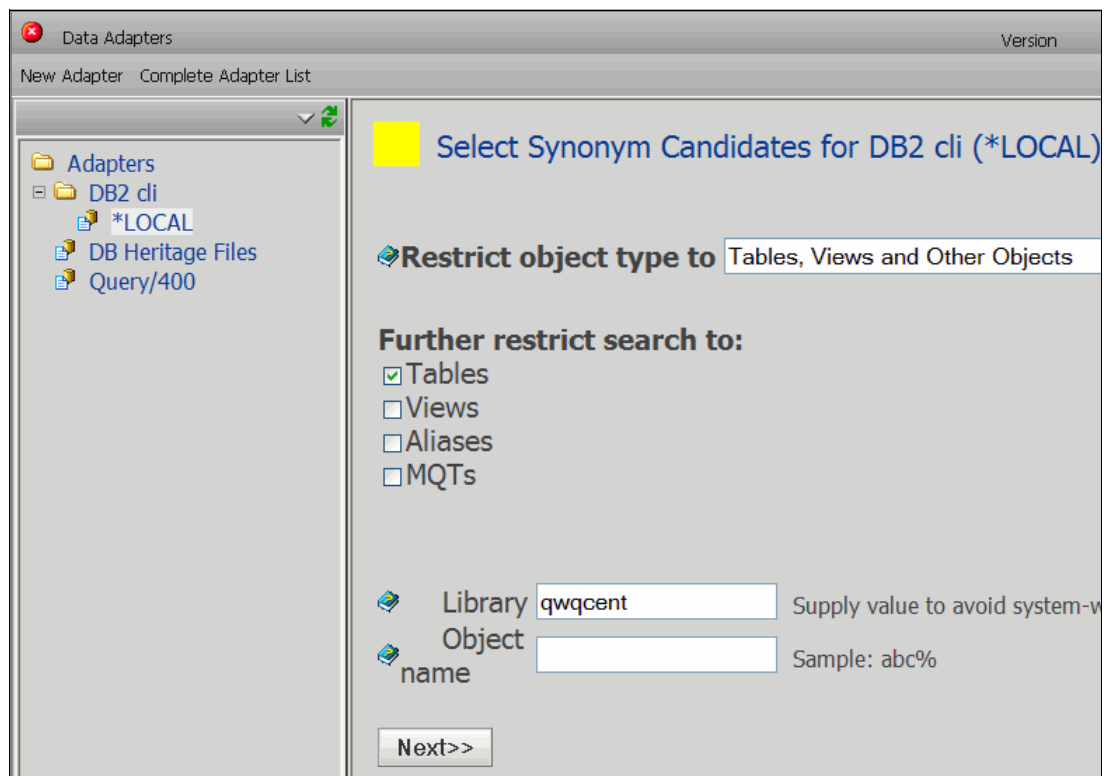


Figure 4-7 Select Synonym Candidates for DB2 cli pane

14. In the Create Synonym for DB2 cli pane (Figure 4-8), complete these steps:
 - a. In the list of tables, select the check box next to **Default synonym name** to select all the tables that are currently displayed.
 - b. For Prefix, enter Cen_. This prefix is added to the beginning of the table names, so the users see, for example, Cen_ALERTS, CEN_ORDERS, and so on. If you have a long list of table names, using a prefix is one way to keep all related tables grouped together.
 - c. Select **With foreign keys** to bring in all tables that are directly related through referential integrity and foreign key support. For more details about foreign key support, see “Benefits of referential integrity for metadata” on page 38.
 - d. Click the **Create synonym** button.

Create Synonym for DB2 cli (*LOCAL) Step 2

☒ With foreign keys ☐ One-part name

Application Prefix Suffix

☒ Overwrite existing synonyms

☒	Default synonym name	Library/Schema	Table name	Type
☒	ALERTS	QWQCENT	ALERTS	TABLE
☒	CURRRATE	QWQCENT	CURRRATE	TABLE
☒	HR	QWQCENT	HR	TABLE
☒	INVENTORY	QWQCENT	INVENTORY	TABLE
☒	ORDERS	QWQCENT	ORDERS	TABLE
☒	PLANT	QWQCENT	PLANT	TABLE
☒	STORES	QWQCENT	STORES	TABLE

Figure 4-8 Creating QWQCENT metadata

You have now authorized your user profile to create tables in the Common Domain. You have created a folder to store all your work in called Tutorials. You have also defined all the tables that we query to DB2 Web Query through the Create Synonym process. See Figure 4-9.

Application	Synonym name	Status
baseapp	Cen_STORES	Created successfully
baseapp	Cen_PLANT	Created successfully
baseapp	Cen_ORDERS	Created successfully
baseapp	Cen_INVENTORY	Created successfully
baseapp	Cen_HR	Created successfully
baseapp	Cen_CURRRATE	Created successfully
baseapp	Cen_ALERTS	Created successfully

Figure 4-9 Synonym creation completion



Report Assistant

Report Assistant provides an intuitive, graphical interface with drag-and-drop functionality. It enables you to quickly create a report in which you can sort data; include record selection criteria; add headings, footings, subtotals, and page breaks; apply a style template; and more. When you use Report Assistant, DB2 Web Query creates a styled report that you can deploy on the Web without learning the complexities of any reporting language. With DB2 Web Query, you can specify input parameters to allow users to easily subset or filter their reports.

Report Assistant supports output in many different file types including PDF files, Microsoft Excel, and DB2 database files. Report Assistant's tight integration with Excel includes the real-time creation of Excel spreadsheets with full styling, drill-downs, and formula capabilities. This support allows Excel power users to analyze their corporate data in a tool with which they are already familiar.

Many of the reports created in this chapter are prerequisites for the enhanced features that are discussed later in this book. Active Reports, which is for users who are not connected to a System i machine, and online analytical processing (OLAP)-enabled reports are both based on existing reports that are created in this chapter.

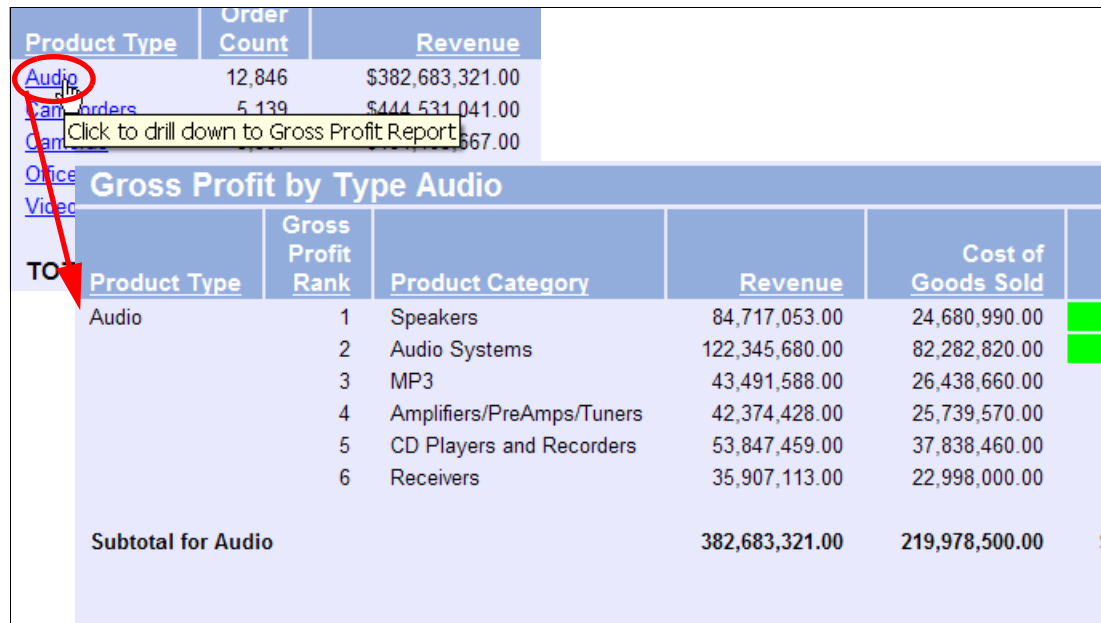
In this chapter, we go into detail about basic functionality that is common among the various DB2 Web Query components. For more details about the various options that are not discussed in the tutorials, use the help text.

Tip: If you have Developer Workbench installed, you will find the help text to be more extensive than the help text that is available in Report and Graph Assistant.

5.1 Tutorial overview

In this tutorial, we create the two reports, which are shown in Figure 5-1. The first report is a simple summary that shows revenue by product type. The second report is a more detailed report that calculates gross profit for each product category within a specific user-requested product type. Finally we enable a user who is viewing the initial summary report to click the product type and automatically drill down to see the more detailed gross profit report.

In 5.5, “Joining tables (reference only)” on page 106, we demonstrate how to join two tables if you did not previously define the relationship using DB2 foreign key support. This section is for your reference only and is not part of the tutorial.



Product Type	Order Count	Revenue
Audio	12,846	\$382,683,321.00
Car Audio	5,139	\$444,531,041.00
Car Stereo		\$667.00

Gross Profit by Type Audio					
Product Type	Gross Profit Rank	Product Category	Revenue	Cost of Goods Sold	
Audio	1	Speakers	84,717,053.00	24,680,990.00	
	2	Audio Systems	122,345,680.00	82,282,820.00	
	3	MP3	43,491,588.00	26,438,660.00	
	4	Amplifiers/PreAmps/Tuners	42,374,428.00	25,739,570.00	
	5	CD Players and Recorders	53,847,459.00	37,838,460.00	
	6	Receivers	35,907,113.00	22,998,000.00	
Subtotal for Audio			382,683,321.00	219,978,500.00	

Figure 5-1 Final drill-down reports

5.2 Report creation

In our first task, we show the product revenue for each product type. We use the Century database that ships with DB2 Web Query in the QWQCENT library. You can find instructions for creating the metadata that defines this database to DB2 Web Query in Chapter 4, “Getting started with the tutorials” on page 57.

Important: In order to complete the tasks in this chapter, you must have read and completed the steps in Chapter 4, “Getting started with the tutorials” on page 57.

In this section, we explain how to create a summary and detailed report. As we explain how to create this report, we illustrate how to perform the following tasks among others:

- ▶ Define a report layout
- ▶ Apply date formatting
- ▶ Use selection criteria
- ▶ Define traffic lighting
- ▶ Generate subtotals and report headings

5.2.1 Creating a summary report (RA1_Revenue)

To begin creating a summary report:

1. In the Reports navigation area, under Domains, expand **Common Domain** → **Reports**. Right-click the **Tutorials** folder, which is where you will store your report, and select **Report Assistant**.

Note: Depending on the number of tables that were defined previously to DB2 Web Query, your list might contain only one table or it might contain many tables.

2. In the window that opens, you see a list of available tables for you to query. Select the **CEN_ORDERS** table. As shown in the example in Figure 5-2, you can tell from the word *Cluster* in the description that this single definition covers multiple tables without requiring external joins. Click **OK**.

Tip: When you type the first character of the table name, you jump to the tables that start with that character. In the “Prefix and suffix (optional, yet recommended)” note box on page 36, we recommend that, when you define your tables (create synonyms) to DB2 Web Query, you add a prefix to the tables so that they are logically grouped, for example by subject area, collection, or library.

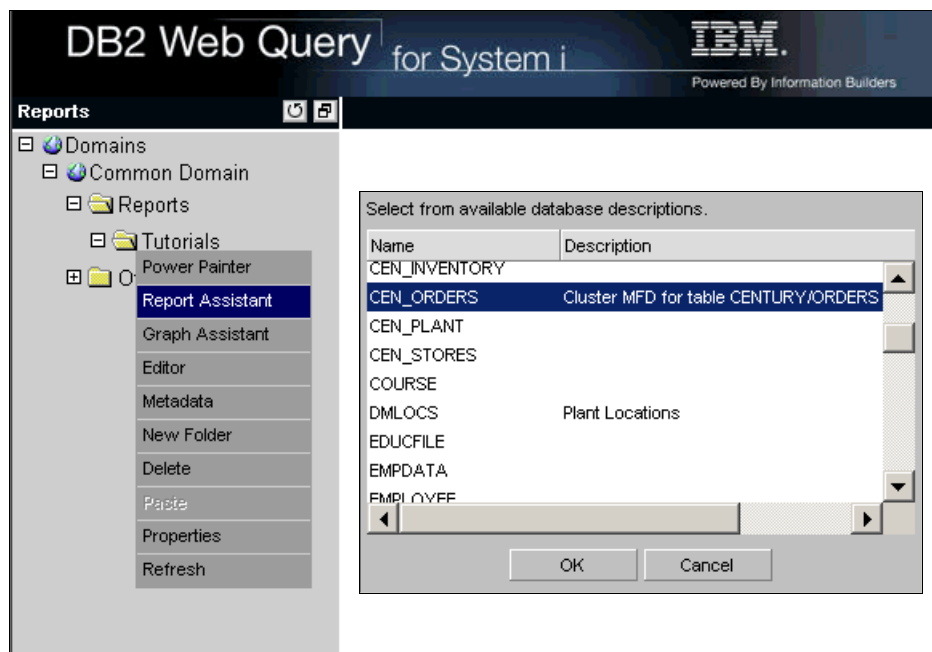


Figure 5-2 Initial table selection using Report Assistant

After you choose the **CEN_ORDERS** table, you see a list of all the column or field names on the Field selection tab. Continue with the next section to learn how to use these names to define a report layout.

5.2.2 Defining a report layout

As discussed in “Benefits of referential integrity for metadata” on page 38, the displayed fields in the Available fields list come from various tables. These tables are automatically brought in

by DB2 Web Query when you select the Foreign key option during the creation of the Orders synonym or metadata. This results in the composite list of available fields.

To define a report layout:

3. To see the table or segment names, drag the sides of the window to widen it.

Icons for Available fields: You can choose to view your field names alphabetically regardless of which table they originated from, or you can view the fields for each table separately. Immediately above the field list are three icons. The first from the left is a calculator icon. The second icon is the dimension or hierarchy icon, which shows the fields within each individual table. The third icon is the list icon, which shows all the fields from all the tables in one long alphabetical list.

4. From the list of Available fields on the left, drag **PRODUCTTYPE** to the Sort by pane in the center of the window.

Tip: If you are using the Firefox browser, you might not be able to drag your fields. In this case, you must highlight the column in which you want to insert in your report and then click the add field icon in the appropriate pane, for example Sort by, Sum, and so on, to where you want to move the field.

5. Again from the list of available fields, drag **LINETOTAL** to the Sum/Print pane on the right side of the window. LINETOTAL is the gross revenue for each line item on an invoice.
6. Select the **Add grand totals to the end of the report** check box. See Figure 5-3.

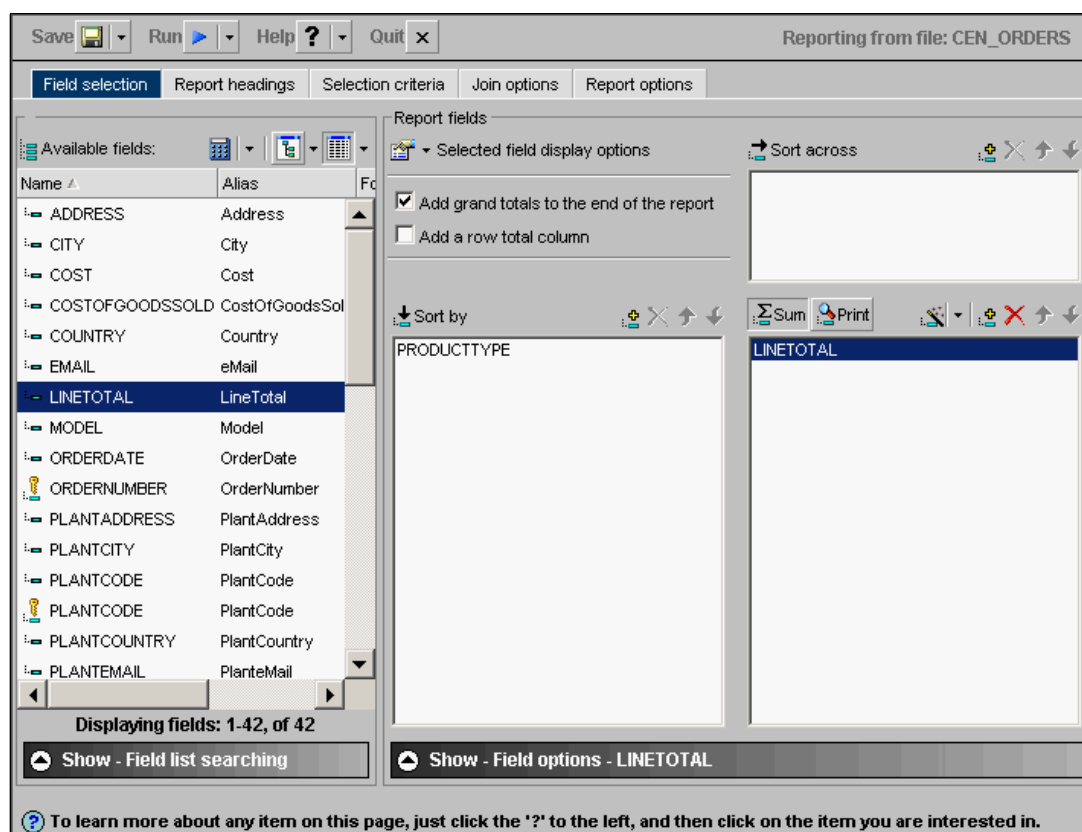


Figure 5-3 Report Assistant field selection

7. At the top of the window, click the **Run** arrow to run your first report. Figure 5-4 shows the initial summary report.

Product Type	Revenue
Audio	382683321.00
Camcorders	444531041.00
Cameras	184103667.00
Office	30245685.00
Video	520360205.00
TOTAL	1561923919.00

Figure 5-4 Initial Summary Report

Tip: If you use an alpha field instead of LINETOTAL, the default sum option shows the value of the alpha field in the last record for each sort group or product type.

8. Save the report as shown in Figure 5-5:
- Click the **Save** button at the top of the window and select **Save As**.
 - In the Save Standard Report window, enter the name as RA1_Revenue and click **OK**.

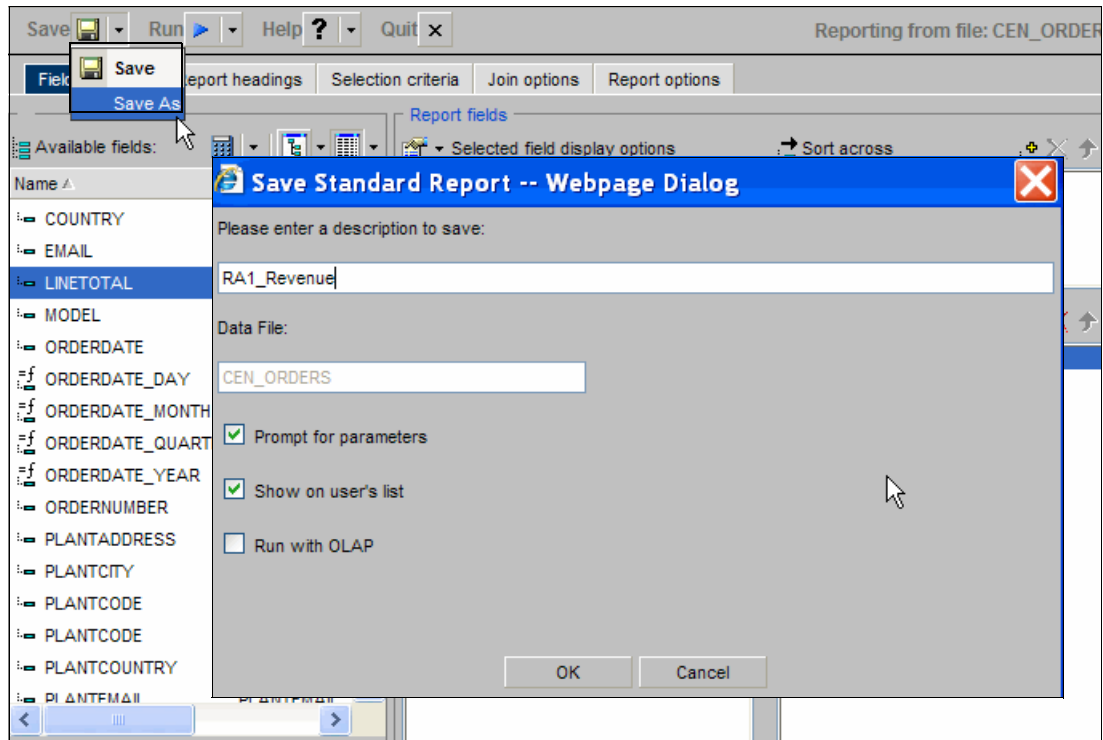


Figure 5-5 Report Assistant save as menu

It was easy to answer the question that was posed about the total revenue by product type, but we know that we can make this report look better and more meaningful for the users. Next we take a closer look at creating sum and detailed reports.

5.2.3 Sum and detail reports

In this section, we look at the difference between the Sum and Print options on the Field selection tab. By default, we click the Sum button, which we used to produce the initial summary report. As requested, this report was aggregated on PRODUCTTYPE. The Print option turns off the aggregation and produces a detail report.

9. Before you run a detail report, limit the number of records that you want to read. There are more than 32,000 detail records in the orders file. We do not need to retrieve all of these records simply to make a point.
 - a. Click the **Report options** tab (Figure 5-6).
 - b. Under Content and generation, for the Limit the number of read operations performed during report generation field, type 100 for the number of records to read.

Figure 5-6 Limiting the records to read on the Report options tab

10. Click the **Field selection** tab.
11. Click **Print** instead of Sum.

12. Click the **Run** arrow at the top of the window. Then you see the report shown in Figure 5-7.

Product Type	Revenue
Audio	75945.00
	17928.00
	46065.00
	249.00
	249.00
	249.00
	33117.00
	249.00
	24651.00

Figure 5-7 Report Assistant detailed print report

13. The Print option is for a detailed report that includes one row for each selected record. The Sum option sums or aggregates field values that are based on your sort column and consolidates the records. Since we want a summary report, complete these steps:

- On the Field selection tab, click **Sum**.
- Click the **Report** options tab.
- On the Report options tab, for the Limit the number of read operations performed during report generation field, clear the value of 100.

14. We now enhance the report to make it more readable:

- Click to highlight the **LINETOTAL** field in the Sum/Print pane.
- Select the **Show - Field options - LINETOTAL** arrow near the bottom of the window.

The Field options display allows you to work with the properties for an individual field. You can describe standard and conditional formats based on values in your report. You can set up a field so that when a user clicks on it they will drill down to a lower level report. You can specify if you want the report to display the actual data in a field or if you want to see the average, minimum, or maximum value in the field instead. If you are working with a sort field you have additional capabilities to control the sort, add ranking columns, and control the subheadings and subfooting.

- i. Click the ellipsis or three dots (...) button under the Format in report as field (Figure 5-8).

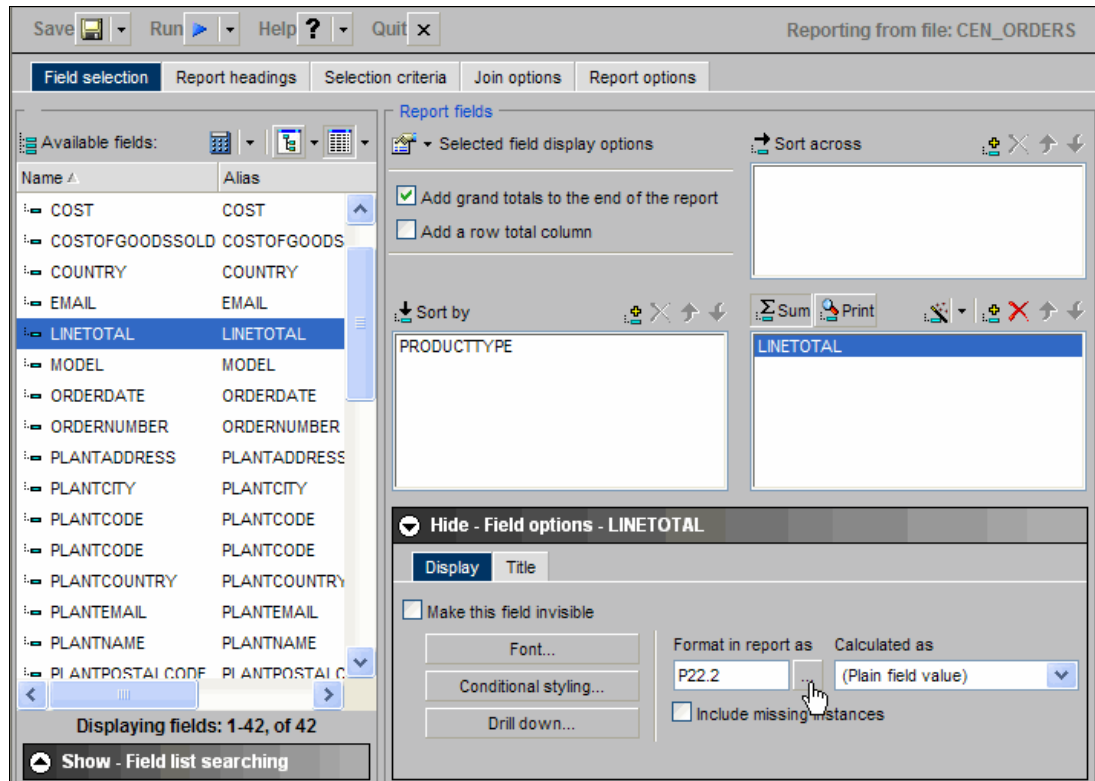


Figure 5-8 Report Assistant field options

- ii. Under Select options (Figure 5-9), select **Commas inclusion** and **Floating dollar** to change the LINETOTAL field to be displayed with commas and a floating dollar sign.

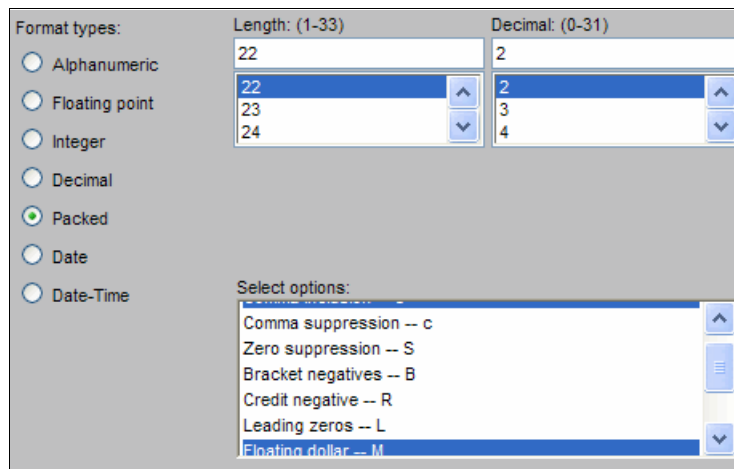


Figure 5-9 Report Assistant change format window

- c. On the Field selection tab, complete these steps:
 - i. Ensure that LINETOTAL is highlighted.
 - ii. Select the **Show - Field options** arrow.
 - iii. Click the **Title** tab. See Figure 5-10. Notice that the title is Revenue. This pane is where you enter your own, more meaningful column headings for users. In our case, the LINETOTAL column was defined to DB2 as having a heading of Revenue. DB2 Web Query imports the DB2 heading attribute into the metadata attribute Title.

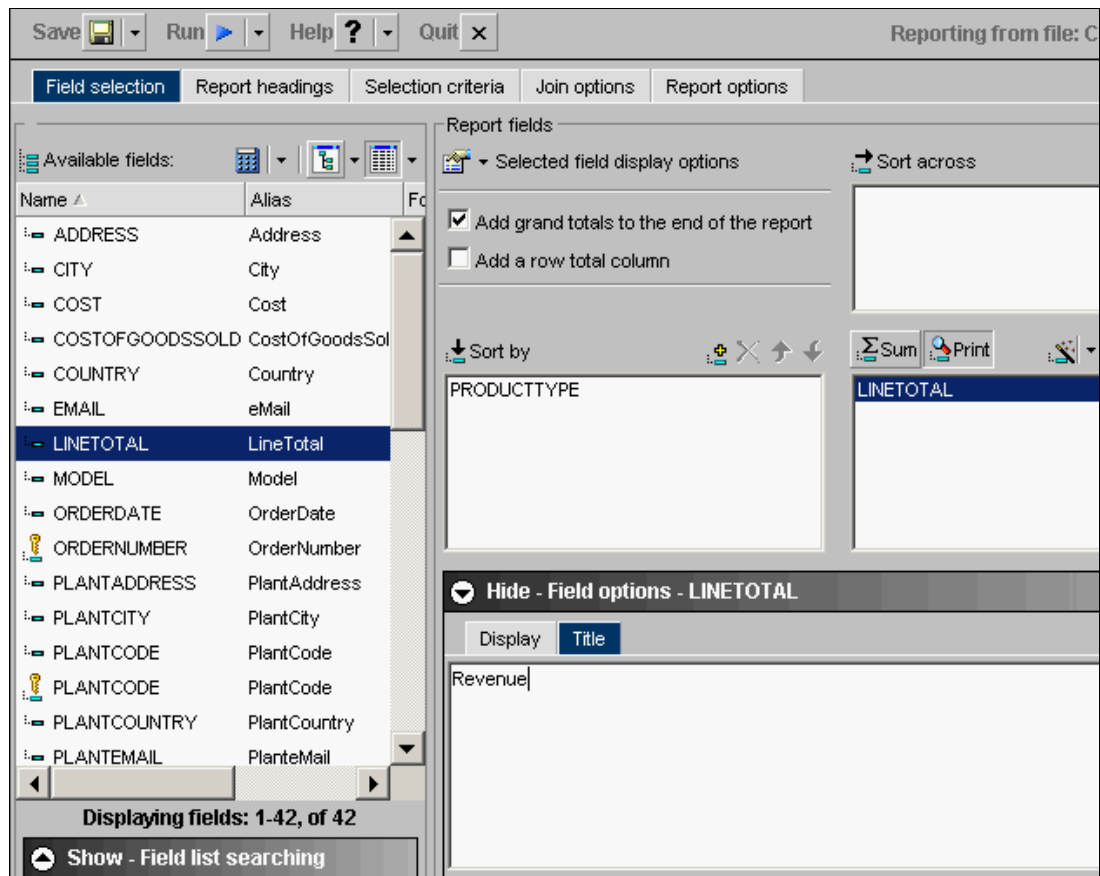


Figure 5-10 Report Assistant new column headings

15. Add a count of the number of orders for each product type:

- a. Drag **ORDERNUMBER** from the Available fields list on the left to the Sum/Print pane on the right.
- b. Click the **Show - Field options** button to open the pane if it is closed.
 - i. Under Calculated as, select **Count**.

Note: If you do not see a long list of field options including Count, verify that your report is a summary report and not a detailed report. Verify that Sum is highlighted, and not Print.

- ii. Under the Format in report as field (Figure 5-11), click the ellipses (...) button. Change the field format to integer seven with commas. The value is displayed in the field as I7C.

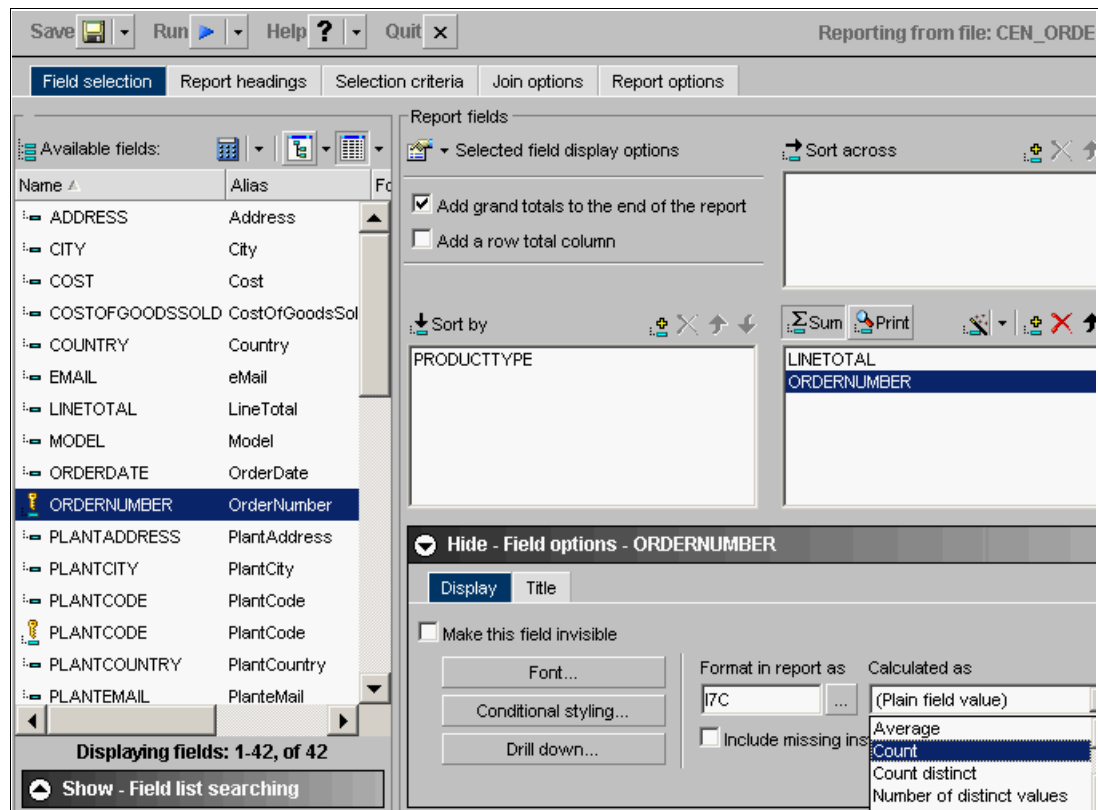


Figure 5-11 Report Assistant field calculation options

- iii. Click the **Title** tab.

- iv. Change the heading as shown in Figure 5-12, with Order on the first line and Count on the second line.

The screenshot shows the 'Report Assistant' window for a report titled 'Reporting from file: CEN_ORDERS'. The window has three tabs: 'Criteria', 'Join options', and 'Report options', with 'Report options' currently selected. Under the 'Report fields' section, there are two checkboxes: 'Add grand totals to the end of the report' (checked) and 'Add a row total column' (unchecked). To the right, there is a 'Sort across' area with a list box containing 'LINETOTAL' and 'CNT.ORDERNUMBER'. Below this, there is a 'Sort by' area with a list box containing 'PRODUCTTYPE'. At the bottom, there is a section titled 'Hide - Field options - ORDERNUMBER' with two buttons: 'Display' and 'Title'. The 'Title' button is highlighted. Below these buttons, there is a list box containing 'Order' and 'Count'.

Figure 5-12 Report Assistant column heading for count field

- c. Using the up and down arrows (circled on the right in Figure 5-13) in the Sum/Print pane, move **CNT.ORDERNUMBER** above LINETOTAL.

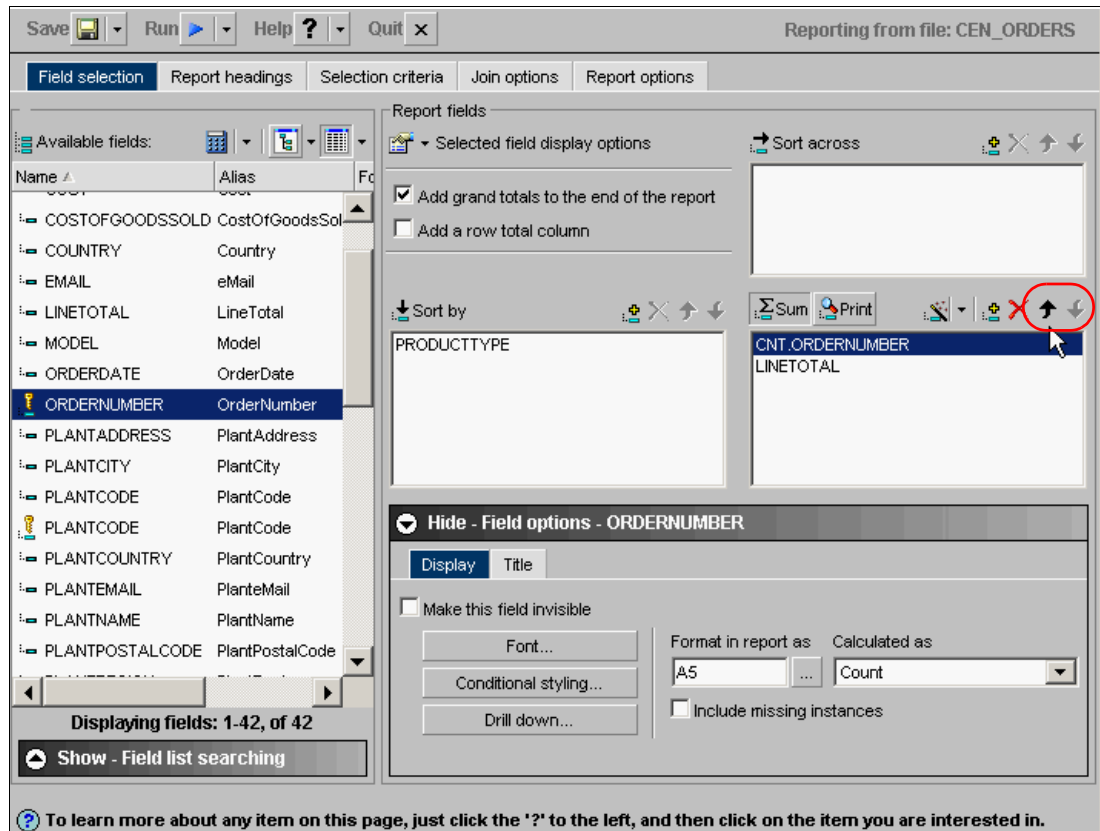


Figure 5-13 Report Assistant field count

16. Click the **Run** arrow at the top of the window to run the report. Figure 5-14 shows the results of running this report. You have now created a report that includes a count of the number of orders for each product type.

Product Type	Order Count	Revenue
Audio	12,846	\$382,683,321.00
Camcorders	5,139	\$444,531,041.00
Cameras	5,307	\$184,103,667.00
Office	2,958	\$30,245,685.00
Video	6,033	\$520,360,205.00
TOTAL	32,283	\$1,561,923,919.00

Figure 5-14 Report Assistant RA1_Revenue report

17. Save your report.

5.2.4 Date formatting (RA2_XTab)

In this next example, we select only products that were sold in the year 2007, and across the top of our report, we add the quarter in which the products were sold.

We create two new fields based on ORDERDATE. One field contains the quarter, and the other field contains the year.

Note: The new quarter and year fields will be available only in the current report. See 12.1.1, “Date Decomposition” on page 269, which explains how to modify the metadata and make these fields available globally. This task requires you to have the optional Developer Workbench feature.

1. If you closed your report, on the DB2 Web Query home page, open the **Tutorials** folder, right-click the **RA1_Revenue** report, and select **Open**.
2. Next to the Field selection list, click the **calculator** icon () to define a new field.
3. In the Define field creator window (Figure 5-15), complete these steps:
 - a. Enter the name of the field as Quarter.
 - b. For Format, select **Q**.
 - c. In the date field, either select or type ORDERDATE. The name must be in uppercase letters. This extracts the quarter data from ORDERDATE.
 - d. Click **OK**.

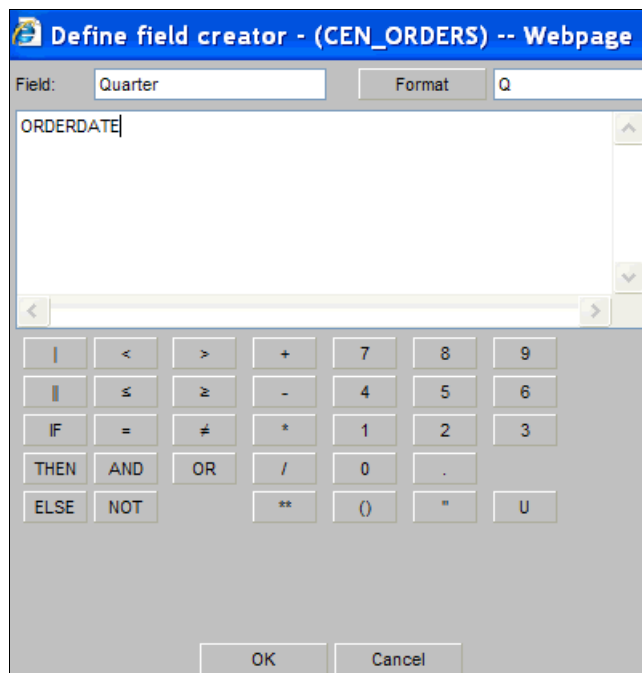


Figure 5-15 Report Assistant defining a date quarter

4. Repeat the process starting with step 2 on page 75 to create the year field. In the Define field creator window (Figure 5-16), for fields, type Year. For Format, select **YY**. In the date field, either select or type ORDERDATE.



Field:	Year	Format	YY
ORDERDATE			

Figure 5-16 Creating a date year field

Note: DB2 Web Query handles dates in an intelligent manner. However, much of this information, including the various date formatting options, is not well described in the help text. For this reason, we provide details about working with dates in “Date and time system variables” on page 368.

5.2.5 Selection criteria

In this section, we continue the procedure by filtering this report and selecting only the year 2007:

5. Click the **Selection criteria** tab.
6. On the Selection criteria tab (Figure 5-17 on page 77), from the Available fields list, drag **Year** to the white pane on the right.

7. From the first drop-down list, select **WHERE**.

WHERE or WHERETOTAL: Although we select the default WHERE, DB2 Web Query allows you to choose WHERE or WHERETOTAL. WHERE does the select at read time and only passes those records to the query that meet the specified condition. WHERETOTAL does the comparison after the records are retrieved, sorted, and aggregated. The selection is done on the aggregate value of the column.

For example, if you want to select only the product types whose total revenue was greater than \$100 million, you use WHERETOTAL. If you use WHERE, you apply the selection on every individual order read and none meet this criteria.

8. Select **EQUAL to**.
9. Click **Select values**.

The screenshot shows the 'Report Assistant Selection criteria' dialog box. It has four tabs: 'Field selection', 'Report headings', 'Selection criteria' (which is active), and 'Join options'. Below the tabs is a 'Report options' button. The 'Available fields' list on the left contains the following items:

Name	Alias
PRODUCTTYPE	ProductType
QUANTITY	Quantity
QUANTITYINSTOCK	QuantityInStock
Quarter	
REGION	Region
RETURNS	Returns
SALESREP	SalesRep
SHIPPINGCOST	ShippingCost
STATE	State
STORECODE	StoreCode
STORECODE	StoreCode
STORENAME	StoreName
TELEFAX	Telefax
TELEPHONE	Telephone
WARRANTYEXP	WarrantyExp
Year	

The 'Screening conditions' section on the right has a 'Delete checked items' button. Below it, a condition is defined for the field 'Year'. The operator is 'EQUAL to'. A '<Select values>' button is next to the operator. A dropdown menu is open, showing 'WHERE' and 'WHERE TOTAL' options.

Figure 5-17 Report Assistant Selection criteria

10. In the window that opens (Figure 5-18), complete these steps:

- a. Click the **Values** radio button.
- b. DB2 Web Query looks for all distinct values in this field and presents them in a list. Select **2007**.

Note: You can compare a field to the following types:

Constant	A specific number
Parameter	A variable whose value or values are either entered by a user or passed from another report
Field	Another field in your table or tables
Values	Selectable from a list of actual values in the selected field

- c. Click **OK**.

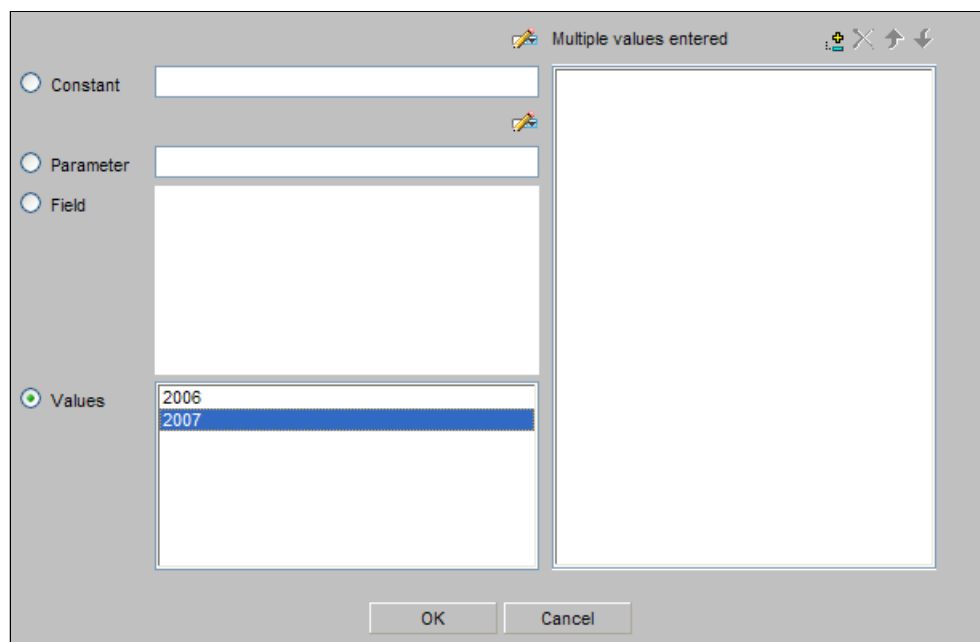


Figure 5-18 Report Assistant selection values

Tip: With DB2 Web Query, you can store a list of products, regions, customers, and so on that you are specifically interested in on your local PC. This list can be stored in a text file or in Excel. If you click the wizard icon (🔧) next to the Constant prompt, you can browse your local PC. The values in that file then become the filter for your query.

5.2.6 Cross-tab report (RA2_XTab)

In this section, we continue our report creation process by creating a cross-tab report:

11. From the Available fields pane, drag **Quarter** to the Sort across pane on the right side of the window (Figure 5-19).

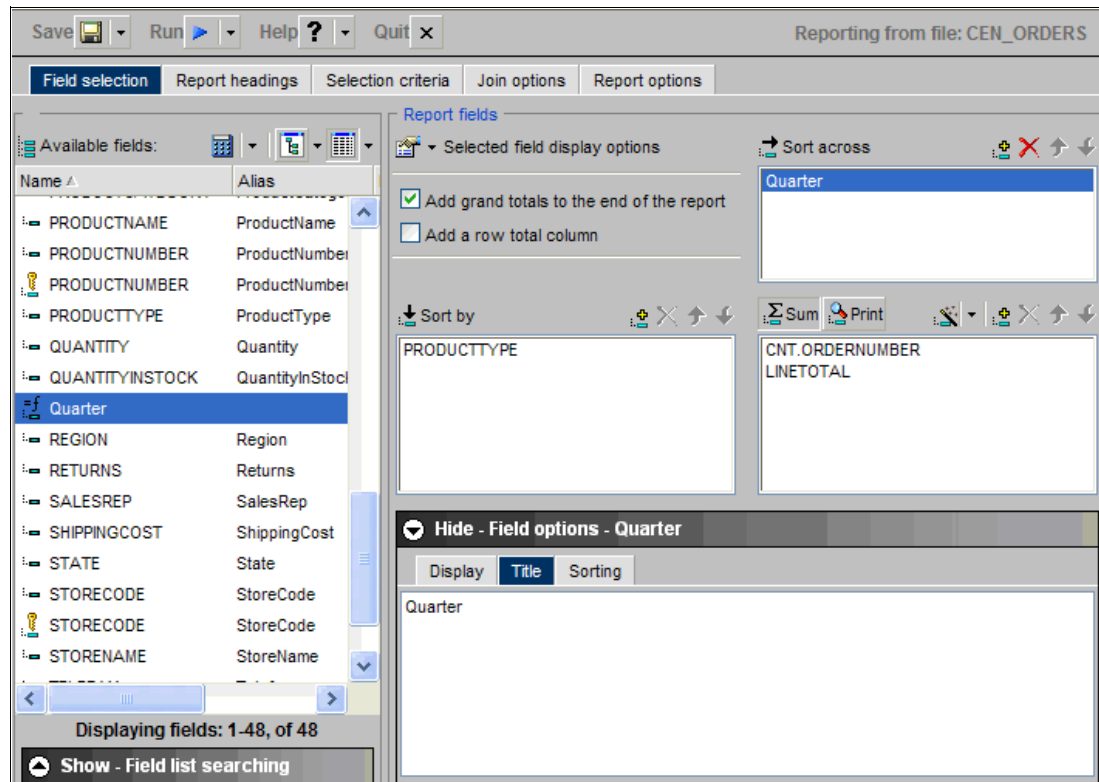


Figure 5-19 Report Assistant Sort across

12. Click the **Run** arrow at the top of the window to run your report. You see the results as shown in Figure 5-20.

Quarter							
Q1		Q2		Q3		Q4	
Product Type	Order Count	Revenue	Order Count	Revenue	Order Count	Revenue	Order Count
Audio	1,782	\$66,724,721.00	2,156	\$55,653,365.00	1,381	\$41,313,395.00	1,029
Camcorders	629	\$68,403,036.00	868	\$66,522,260.00	584	\$51,665,406.00	492
Cameras	669	\$28,003,505.00	880	\$28,006,440.00	594	\$21,686,530.00	450
Office	416	\$5,635,513.00	483	\$4,550,333.00	347	\$3,644,735.00	221
Video	784	\$85,195,987.00	1,043	\$75,051,436.00	663	\$62,214,814.00	527
TOTAL	4,280	\$253,962,762.00	5,430	\$229,783,834.00	3,569	\$180,524,880.00	2,719

Figure 5-20 Report Assistant sort across output

Tip: If your report shows only two quarters, remember that you must remove the 100 record limit. See step 13 on page 69.

13. At the top of the window, click **Save** → **Save As**. In the Save Standard Report window, save the report with the name RA2_XTab, click **OK**, and close it.

5.2.7 Creating a sample detail report (RA3_GP)

In this section, we create a parameterized report where the user selects, at run time, the product type that they want to query. We are also able to drill directly to this report from the RA1_Revenue report that we initially created.

1. Right-click the **Tutorials** folder and choose **Report Assistant**.
2. In the window that opens (Figure 5-21), select the **CEN_ORDERS** table and click **OK**.

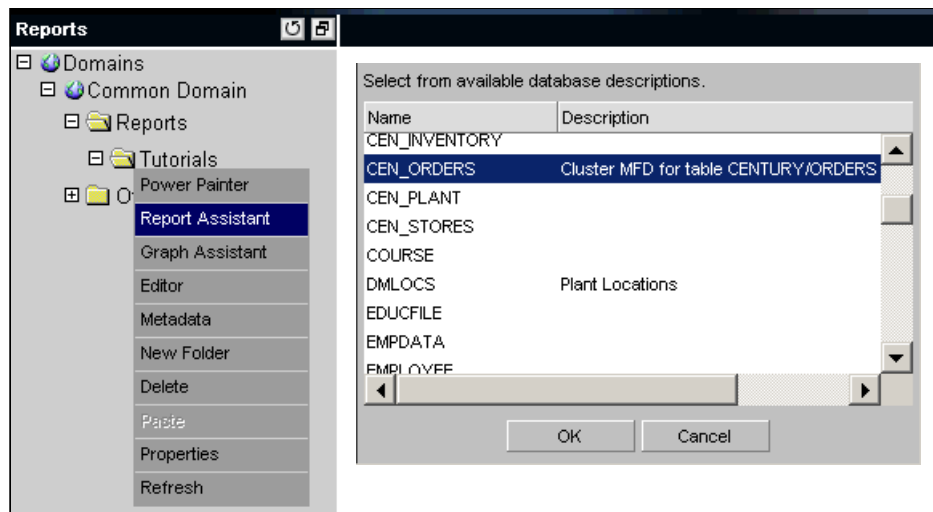


Figure 5-21 Report Assistant file selection

3. From the Available fields pane, drag **PRODUCTTYPE** and **PRODUCTCATEGORY** to the Sort by pane (Figure 5-22).
4. Drag **LINETOTAL** and **COSTOFGOODSSOLD** to the Sum/Print pane.
5. Under Hide - Field options, for Format in report as, click the ellipsis (...) button. Add **commas** to the formatting of LINETOTAL and COSTOFGOODSSOLD.
6. Select the **Add grand totals to the end of the report** check box.

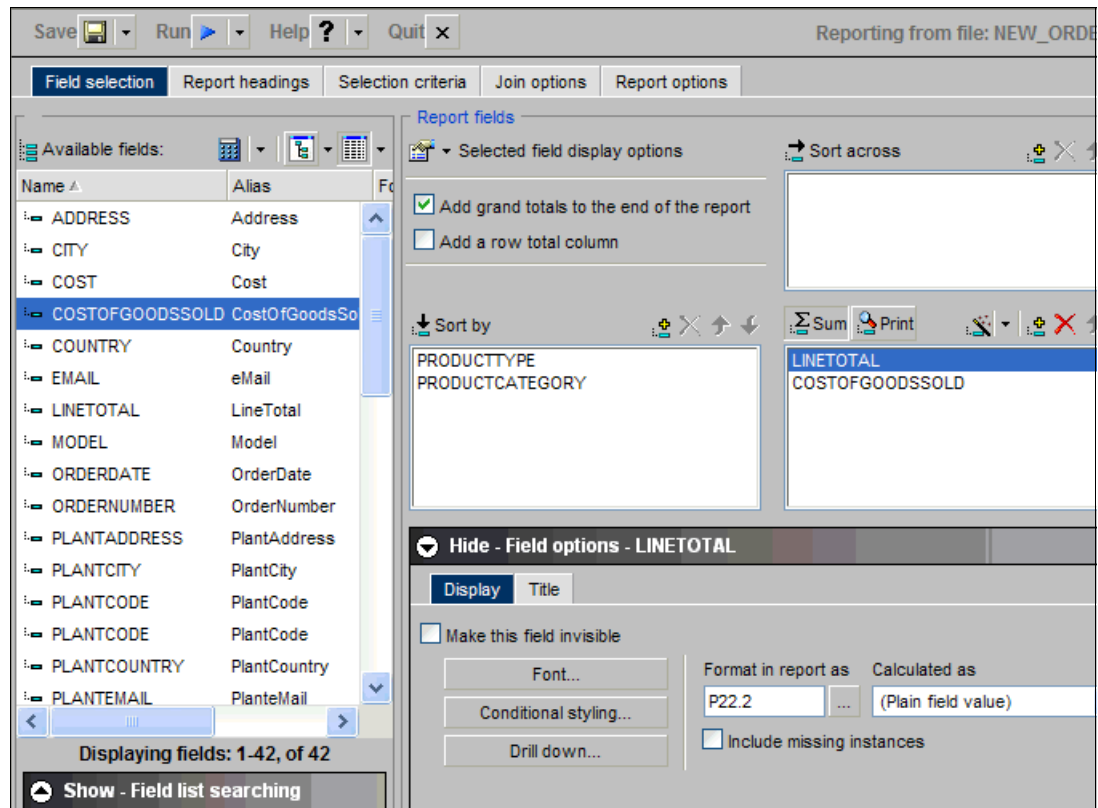


Figure 5-22 Report Assistant Field selection for detail report

Defining and computing fields

In this section, we create new fields and explain how to create new calculated fields in our report:

7. Click the **calculator** icon () , which is next to the Available fields heading, and select **New define field** (Figure 5-23 on page 82). We create a gross profit field by subtracting the COSTOFGOODSSOLD from LINETOTAL.

Note: There are two ways to create calculated fields in DB2 Web Query. When you use the *define* field, you add the definition to the list of fields within the table. This field is calculated every time a record is read and selected. Next to the Sum/Print heading is the wizard wand icon. You can use this icon to create a *compute* field. In this case, the field is not calculated until after the data is sorted and all aggregation is complete. Compute fields are often required for percentages and variances.

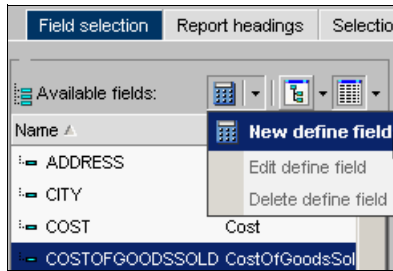


Figure 5-23 Report Assistant New define field

8. In the Define field creator window (Figure 5-24), complete the following steps:
 - a. Enter a new field name of `Gross_Profit`.
 - b. There are three icons on the right side:   . The first two icons display the list of fields in various formats including alphabetically and hierarchically. The function icon is used to display the precoded functions that are available. Try the different buttons.
Click the **list** icon (). This icon shows the list of available field names alphabetically.
 - c. Create the formula `LINETOTAL - COSTOFGOODSSOLD`. Double-click the field **LINETOTAL**, click the minus button (-) on the calculator pad or on your keyboard, and then double-click **COSTOFGOOD**.

Tip: Field names in the Defined field creator panel must all be in uppercase.

- d. Click the **Format** button and define your field as being an integer, 10 long, with commas and a floating dollar sign. It is displayed in the field as `I10CM`.

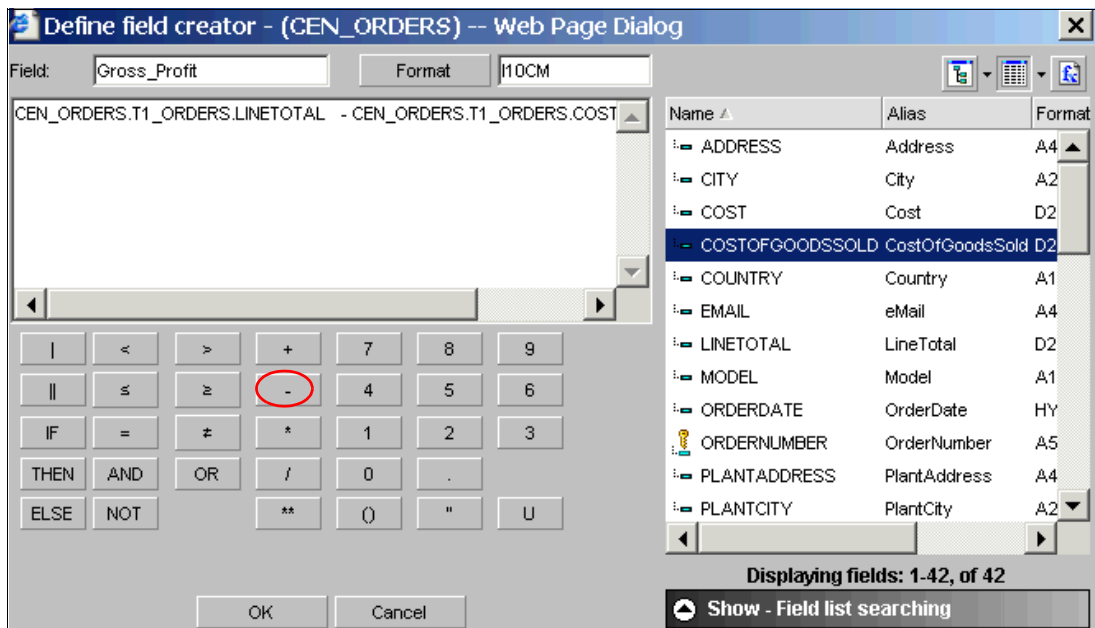


Figure 5-24 Report Assistant creating a defined field

Tip: When creating your own reports, ensure that the field length is large enough to handle the commas, dollar sign, and maximum digits possible.

- e. Click **OK**.
9. Drag your new **Gross_Profit** field to the Sum/Print pane (Figure 5-25). Verify that the Gross_Profit field is highlighted.

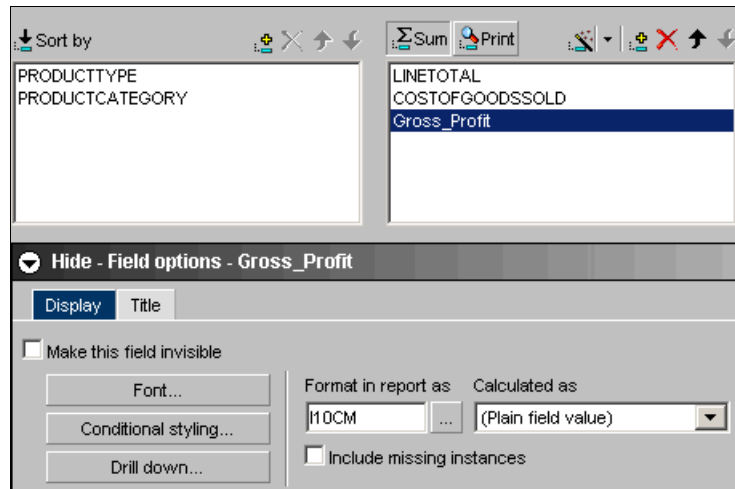


Figure 5-25 Report Assistant formatting gross profit

Tip: There are times when you might create one define field based on a second define field. DB2 Web Query processes the defines in the order in which they are entered into the system. The define field must exist before it can be used in a secondary calculation. For more details about viewing the define order of the fields, see Appendix C, “Processing differences between the define and compute fields” on page 375.

10. Click the **Show - Field options** arrow near the bottom of the window.
 - a. Select the **Title** tab.
 - b. Change the column as shown in Figure 5-26, with Gross on the first line and Profit on the second line.

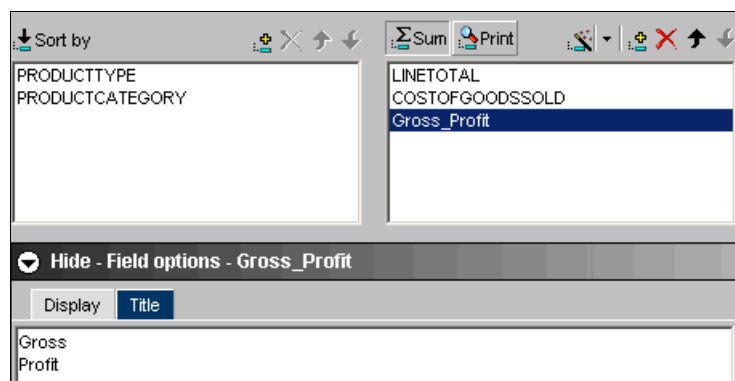


Figure 5-26 Report Assistant Show - Field options

If you define your DB2 tables and specify long names, aliases, or column headings, DB2 Web Query uses them.

Tip: The IF THEN ELSE constructs allow you to use logic when creating virtual fields. For example, if you want to create your own count of the number of items sold in 2007, you can create a virtual field and use the formula:

IF Year EQ 2007 THEN 1 ELSE 0

DB2 Web Query prefers using “EQ” instead of “=”.

5.2.8 Conditional styling (traffic lighting)

Next we highlight the gross profits that are \$25,000,000 or more:

11. In the Sum pane, highlight **Gross_Profit** and open the **Field options** pane.

12. Select the **Display** tab (Figure 5-27):

- a. Click the **Conditional styling** button.

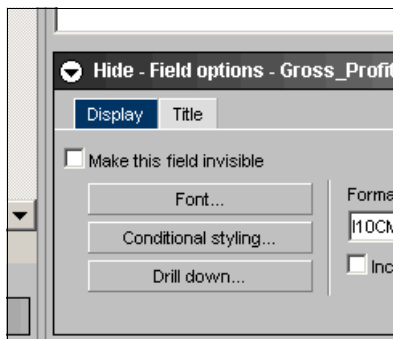


Figure 5-27 Report Assistant Conditional styling option

- b. In the Conditional styling window (Figure 5-28), click the **wizard** icon.

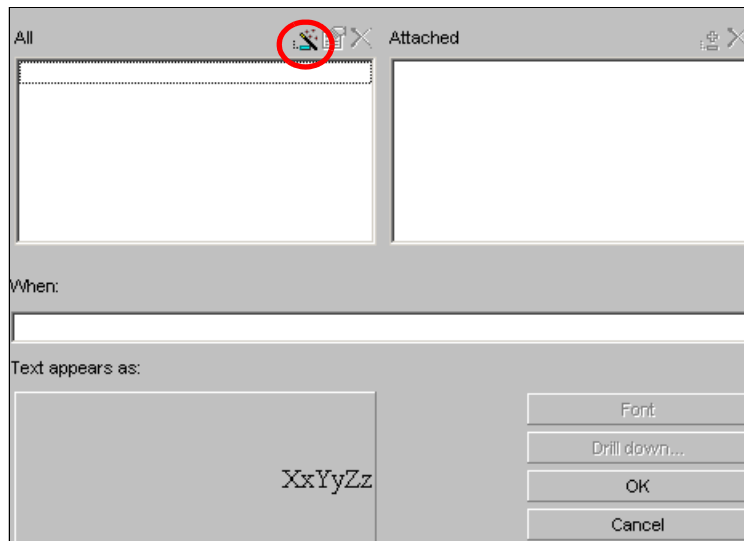


Figure 5-28 Report Assistant Conditional styling window

- c. In the Edit condition window (Figure 5-29), you can define various filters or conditions for which you want to test your data:
 - i. Enter the name `High_Profit` for this condition.
 - ii. For Field, select your test field of **Gross_Profit**.
 - iii. For Relations, select the **is greater than or equal to** relationship.
 - iv. Under Please select a value, enter a value of 25000000 (25 million).
 - v. Click **OK**.

Values button: The Conditional styling dialog allows you to compare `Gross_Profit` to another field in your table. If you click the Values button, you see a list of distinct values in the `Gross_Profit` column.

Figure 5-29 Report Assistant edit condition window

You can now enter another condition. For example, you might want to create a condition called Low Profit. This allows you to monitor your report and present the high and low in different colors.

Note: Now that you have created a condition, you must “attach” it to your column for it to take effect. You can create one condition and attach it to multiple columns. For example, in a high profit situation, you might want every column in the report to be highlighted.

- d. From the list of conditions, highlight the one in which your are interested, **High_Profit**, and then click the **add** icon on the right side of the window to attach the condition to the `Gross_Profit` column. That was the column that we started with when we clicked the Conditional styling button (Figure 5-30).

Figure 5-30 Report Assistant attach conditional styling

13. Define the formatting of the Gross_Profit field when it meets the High_Profit condition.
 - a. Click the **Font** button.
 - b. To make the profit stand out, highlight it by choosing the **Bold** style and **green** for the background color (Figure 5-31). Click **OK** twice to return to the main reporting window.

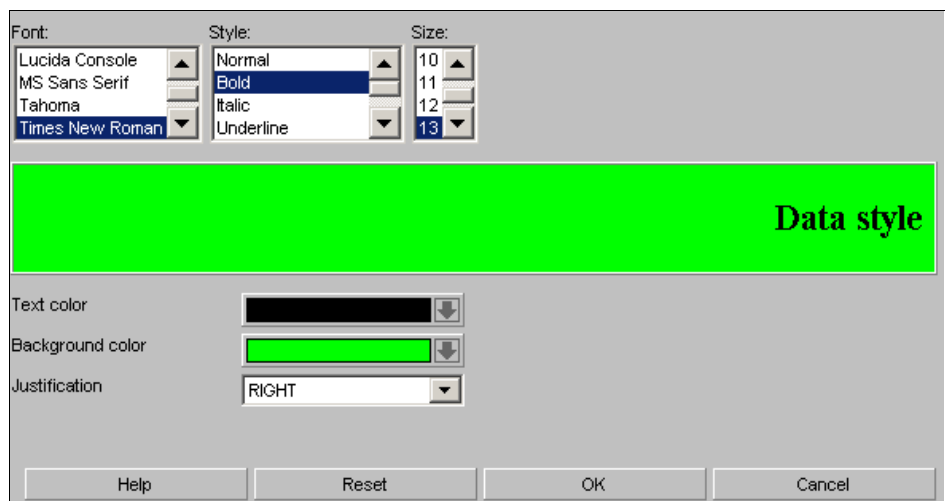


Figure 5-31 Report Assistant styles window

5.2.9 Using variables in report headings and footings

In this section, we explain how to enhance our report by using variables in the report headings and in page footing:

14. Click the **Report headings** tab.
15. Under Page heading, enter Gross Profit by Product Category. For Page footing, type Prepared on &DATE at &TOD (Figure 5-32).

Tip: For more information about using date and time variables within DB2 Web Query, see “Date and time system variables” on page 368.



Figure 5-32 Report Assistant page headings and footings

Note: DB2 Web Query ships with a default style sheet that automatically formats headings and footings. To control the report formatting yourself, you must remove the default style sheet under the Report options tab.

16. Click **Run** at the top of the window to run your report. Figure 5-33 shows the resulting report. Notice how easy it is to see the high profit items that are highlighted in green.

Gross Profit by Product Category				
Product Type	Product Category	Revenue	Cost of Goods Sold	Gross Profit
Audio	Amplifiers/PreAmps/Tuners	42,374,428.00	25,739,570.00	\$16,634,858
	Audio Systems	122,345,680.00	82,282,820.00	\$40,062,860
	CD Players and Recorders	53,847,459.00	37,838,460.00	\$16,008,999
	MP3	43,491,588.00	26,438,660.00	\$17,052,928
	Receivers	35,907,113.00	22,998,000.00	\$12,909,113
	Speakers	84,717,053.00	24,680,990.00	\$60,036,063
Camcorders	Digital8 Camcorders	13,614,953.00	6,512,600.00	\$7,102,353
	DVD Camcorders	379,376,637.00	300,373,350.00	\$79,003,287
	MiniDV Camcorders	51,539,451.00	34,128,360.00	\$17,411,091
Cameras	Digital Cameras	184,103,667.00	133,328,830.00	\$50,774,837
Office	Handheld and PDA	18,533,190.00	14,067,420.00	\$4,465,770
	Organizers	11,712,495.00	4,957,305.00	\$6,755,190
Video	DVD	329,872,045.00	248,768,900.00	\$81,103,145
	TV	168,799,539.00	150,771,700.00	\$18,027,839
	VCR	21,688,621.00	16,270,950.00	\$5,417,671
TOTAL		1,561,923,919.00	1,129,157,915.00	\$432,766,004
Prepared on 09/16/07 at 19.19.45				

Figure 5-33 Report Assistant high profit report

17. Save your report. Click the **Save** button at the top of the window and select **Save As**. In the Save Standard Report window, enter the name it RA3_GrossPft and click **OK**.

5.2.10 Generating subtotals

We continue our example by generating subtotals for each product type:

18. In the Sort by field, highlight **PRODUCTTYPE**.

19. Make sure that the **Show - Field options** pane (Figure 5-34) is displayed.

- Click the **Sorting** tab.
- Select **Subtotal numeric sum/print fields**.

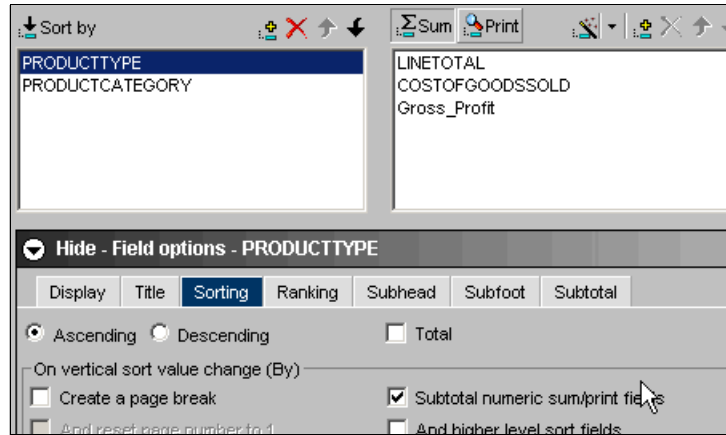


Figure 5-34 Report Assistant subtotals

20. Click the **Run** arrow at the top of the window to run your report again. Figure 5-35 shows the results of running this report.

Gross Profit by Product Category				
Product Type	Product Category	Revenue	Cost of Goods Sold	Gross Profit
Audio	Amplifiers/PreAmps/Tuners	42,374,428.00	25,739,570.00	\$16,634,858
	Audio Systems	122,345,680.00	82,282,820.00	\$40,062,860
	CD Players and Recorders	53,847,459.00	37,838,460.00	\$16,008,999
	MP3	43,491,588.00	26,438,660.00	\$17,052,928
	Receivers	35,907,113.00	22,998,000.00	\$12,909,113
	Speakers	84,717,053.00	24,680,990.00	\$60,036,063
*TOTAL Audio		382,683,321.00	219,978,500.00	\$162,704,821
Camcorders	Digital8 Camcorders	13,614,953.00	6,512,600.00	\$7,102,353
	DVD Camcorders	379,376,637.00	300,373,350.00	\$79,003,287

Figure 5-35 Report Assistant subtalled report

21. To change the default subtotal heading, make sure that **PRODUCTTYPE** is highlighted.
22. In the Field options pane, click the new **Subtotal** tab (Figure 5-36).
23. Under Current Subtotal Text, type Subtotal for. Make sure that you leave a blank at the end because DB2 Web Query appends the product type to your heading.

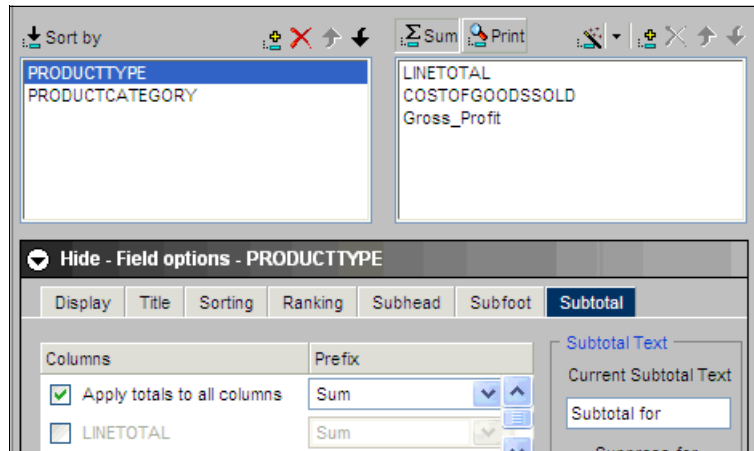


Figure 5-36 Report Assistant new subtotal headings

Tip: By default, all columns are summed to create a total line. You can override individual columns and request that, instead of summing the column, the total line should show the result of calculations such as a count of values or the average of the values. Look at the options under Sum.

24. Click the **Run** arrow to run the report. Figure 5-37 shows the results of running the report.

Gross Profit by Product Category				
Product Type	Product Category	Revenue	Cost of Goods Sold	Gross Profit
Audio	Amplifiers/PreAmps/Tuners	42,374,428.00	25,739,570.00	\$16,634,858
	Audio Systems	122,345,680.00	82,282,820.00	\$40,062,860
	CD Players and Recorders	53,847,459.00	37,838,460.00	\$16,008,999
	MP3	43,491,588.00	26,438,660.00	\$17,052,928
	Receivers	35,907,113.00	22,998,000.00	\$12,909,113
	Speakers	84,717,053.00	24,680,990.00	\$60,036,063
Subtotal for Audio		382,683,321.00	219,978,500.00	\$162,704,821
Camcorders	Digital8 Camcorders	13,614,953.00	6,512,600.00	\$7,102,353
	DVD Camcorders	270,276,627.00	200,272,250.00	\$70,004,377

Figure 5-37 Report Assistant report showing new subtotal headings

5.2.11 Sorting by an aggregate field

Often when users see the results of what they requested, they want something different. Instead of a product category within a product type sequence, the user now requests that we sort by gross profit within product type:

25. The Show - Field options pane for Gross_Profit does not contain a Sort tab. Drag **Gross_Profit** from the Available fields pane to the Sort by pane. Then move it below PRODUCTTYPE as shown in Figure 5-38.
26. In the Show - Field options pane, click the **Sorting** tab and select the **Total** option. If you do not perform these steps, you will work with individual gross profit records instead of the aggregate or total of gross profit by product type.

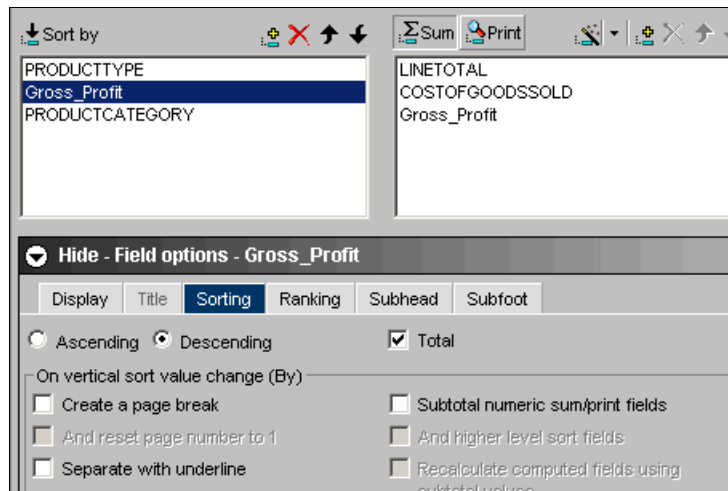


Figure 5-38 Report Assistant adding additional sort criteria

Tip: If you run your report without selecting the Total option, you see that the report contains detail records instead of summary records. Therefore, it is important to remember to select the Total option.

27. If you do nothing else, your report contains the Gross_Profit column twice. Select the **Display** tab and click the **Make this field invisible** option to make Gross_Profit invisible (Figure 5-39).

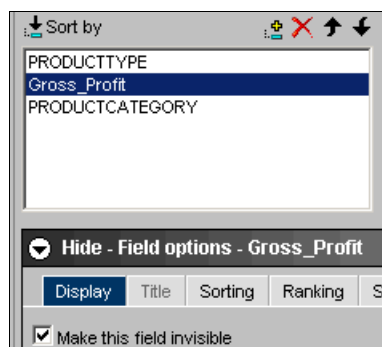


Figure 5-39 Report Assistant make field invisible

28. Click the **Sorting** tab and select the **Descending** sequence.

29. Save your report.

30. Click the **Run** arrow to run the report. Figure 5-40 shows the results of running the report.

Notice that the records within each product type are in descending sequence by gross profit instead of product category.

Product Type	Product Category	Revenue	Cost of Goods Sold	Gross Profit
Audio	Speakers	84,717,053.00	24,680,990.00	\$60,036,063
	Audio Systems	122,345,680.00	82,282,820.00	\$40,062,860
	MP3	43,491,588.00	26,438,660.00	\$17,052,928
	Amplifiers/PreAmps/Tuners	42,374,428.00	25,739,570.00	\$16,634,858
	CD Players and Recorders	53,847,459.00	37,838,460.00	\$16,008,999
	Receivers	35,907,113.00	22,998,000.00	\$12,909,113
Subtotal for Audio		382,683,321.00	219,978,500.00	\$162,704,821
Camcorders	DVD Camcorders	379,376,637.00	300,373,350.00	\$79,003,287
	MiniDV Camcorders	51,539,451.00	34,128,360.00	\$17,411,091

Figure 5-40 Report Assistant sorting by a sum field

5.2.12 Adding ranking columns (RA4_Advanced)

We now show the gross profit ranking within the product type:

31. In the Sort by pane, highlight **Gross_Profit**.

32. Click **Show - Field options** if it is not in view.

33. Click the **Ranking** tab.

34. On the Ranking tab, select the **Add Ranking column**. Change the column as shown in Figure 5-41, with Gross on the first line, Profit on the second line, and Rank on the third line.

The screenshot shows the 'Sort by' pane in the Report Assistant. The 'Sort by' list contains 'PRODUCTTYPE', 'Gross_Profit' (highlighted), and 'PRODUCTCATEGORY'. Below this, the 'Hide - Field options - Gross_Profit' section is expanded, showing the 'Ranking' tab. The 'Add Ranking column' checkbox is checked. Below it, there is a checkbox for 'Limit the number of ranked values to' with a value of 5. At the bottom, the ranking criteria are listed as 'Gross', 'Profit', and 'Rank'.

Figure 5-41 Report Assistant ranking

35. Save your report. Click the **Save** button at the top of the window and select **Save As**. In the Save Standard Report window, enter the name it RA4_Advanced and click **OK**.

36. Click the **Run** arrow at the top of the window to run your report. Figure 5-42 shows the results of running the report.

Tip: If you want to produce a “Top 10” report, use the ranking capability. From the Ranking tab, you limit your ranked values to 10.

Product Type	Gross Profit Rank	Product Category	Revenue	Cost of Goods Sold	Gross Profit
Audio	1	Speakers	84,717,053.00	24,680,990.00	\$60,036,063
	2	Audio Systems	122,345,680.00	82,282,820.00	\$40,062,860
	3	MP3	43,491,588.00	26,438,660.00	\$17,052,928
	4	Amplifiers/PreAmps/Tuners	42,374,428.00	25,739,570.00	\$16,634,858
	5	CD Players and Recorders	53,847,459.00	37,838,460.00	\$16,008,999
	6	Receivers	35,907,113.00	22,998,000.00	\$12,909,113
Subtotal for Audio			382,683,321.00	219,978,500.00	\$162,704,821
Camcorders	1	DVD Camcorders	379,376,637.00	300,373,350.00	\$79,003,287
	2	MiniDV Camcorders	51,539,451.00	34,128,360.00	\$17,411,091

Figure 5-42 Report Assistant with a ranking column

5.3 Output options

In this section, we experiment with a few of the other Report options within DB2 Web Query. The first one is on-demand paging.

To continue with this section, the RA4_Advanced report must be open. If you must open this report, on the DB2 Web Query home page, right-click the **RA4_Advanced** report and select **Open**.

5.3.1 On-demand paging

Choose the **Report options** tab. On this tab, select the **On-demand paging** option. Then rerun your report. See Figure 5-43.

Field selection | Report headings | Selection criteria | Join options | **Report options**

Generate report as: HTML - Web browser (with grid)

Output format specific

Report title:

☒ Optimize output with Cascading Style Sheets

☐ Use accessibility features (Section 508)

Summary text for report:

Interactive options

Enable OLAP: Off Automatic drill down: Off

☒ On-demand paging

☐ Show pop-up field description

Figure 5-43 Report Assistant On-demand paging option

Our sample report is not the best to show the on-demand paging feature, but you can see the options that are available with on-demand paging as shown in Figure 5-44. Assume that our report was 201 pages long instead of one page long. When on-demand paging is enabled, DB2 Web Query saves your report on the System i machine and delivers one page of report output at a time to your browser, decreasing the amount of time that you wait for your report to be transmitted. The bulk of your report remains on the System i machine until you request it or close the report. On-demand paging allows you to jump directly to a specific page or to search for a string of information in the report.

Video	1	DVD	329,872,045.00	248,768,900.00	\$81,103,145
	2	TV	168,799,539.00	150,771,700.00	\$18,027,839
	3	VCR	21,688,621.00	16,270,950.00	\$5,417,671
Subtotal for Video			520,360,205.00	415,811,550.00	\$104,548,655
TOTAL			1,561,923,919.00	1,129,157,915.00	\$432,766,004

Page 1 of 1

Jump Next page Search

Figure 5-44 Report Assistant On-demand paging output

5.3.2 PDF output

Up to this point, all of our output has been to a browser. DB2 Web Query provides many additional options for output. In this section, we look at the PDF output option.

Note: To create a PDF report, no additional software is required other than the no-charge Adobe Reader®. Adobe Acrobat Professional Edition is also not required.

Click the **Report options** tab. For Generate report as, select **PDF - Portable Document Format** (Figure 5-45). Then click the **Run** arrow to generate the report again.

Field selection | Report headings | Selection criteria | Join options | **Report options**

Generate report as: HTML - Web browser

Output format specified: HTML - Web browser, HTML - Web browser (with grid), AHTML - Active Report, **PDF - Portable Document Format**, EXL97 - Excel 97, EXL2K - Excel 2000, EXL2KFORMULA - Excel w/ formula, PS - PostScript

Report title:

☒ Optimize output

☐ Use accessibility

Show styled formats:

Content and generation

Page numbering: Default

Limit the number of read of report generation:

Figure 5-45 Report Assistant PDF report options

Restriction: The report shown in Figure 5-46 was generated using an imported style sheet. If your PDF report scrolls onto two pages, you run into a known problem with the default style sheet. This problem is being investigated as this book goes to print. Not all style sheets have this problem.

See “I have my own style sheets that I want to use for my reports. How do I make them available from Report Assistant?” on page 351. This FAQ includes information about how to import style sheets. The report in Figure 5-46 was created using the “problue.sty” style sheet that ships with DB2 Web Query.

PAGE 1

Gross Profit by Product Category

Product Type	Gross Profit Rank	Product Category	Revenue	Cost of Goods Sold	Gross Profit
Audio	1	Speakers	84,717,053.00	24,880,990.00	\$60,036,063
	2	Audio Systems	122,345,680.00	82,282,820.00	\$40,062,860
	3	MP3	43,491,588.00	26,438,660.00	\$17,052,928
	4	Amplifiers/PreAmps/Tuners	42,374,428.00	25,739,570.00	\$16,634,858
	5	CD Players and Recorders	53,847,459.00	37,838,460.00	\$16,008,999
	6	Receivers	35,907,113.00	22,998,000.00	\$12,909,113
Subtotal for Audio			382,683,321.00	219,978,500.00	\$162,704,821
Camcorders	1	DVD Camcorders	379,376,837.00	300,373,350.00	\$79,003,487
	2	MiniDV Camcorders	51,539,451.00	34,128,360.00	\$17,411,091

Figure 5-46 Report Assistant PDF output

5.3.3 Excel output

There are multiple choices for outputting to Excel. An excellent option is to download your report to Excel, complete with formulas. With this choice, your totals and subtotals, as well as many compute fields, are represented as Excel formulas and are not downloaded as static values. This allows these columns to continue to be accurate even after the user modifies the worksheet.

Excel 97 versus Excel 2007: DB2 Web Query has the ability to create reports as Excel files. These can be created in two formats, Excel binary (XLS) files, referred to as *Excel 97*, and the newer XML-based Excel format (XHT), referred to as *Excel 2000*. Excel 2000 is really the first release that supported XML. DB2 Web Query cannot output directly in the new format of Excel 2007. However, Excel 2007 can read the Excel 2000 format.

With the XML-based Excel 2000, style sheets can be applied, cells can be formatted, traffic lighting or conditional styling can be passed, calculations can be rendered as formulas, and drill-downs are supported from within the resulting spreadsheet.

You should still be in report RA4_Advanced. To generate output in Excel:

1. Click the **Report options** tab.
2. Select **Excel w/formula** from the first drop-down list.

Note: There are many more output choices. Changing the second drop-down list changes the output choices in the first list. If you are not using a style sheet for your output, the difference between HTML and HTML with grid is whether your reports shows with grid lines between the rows and columns. This option is overridden by your style sheet.

3. Run your report.

Figure 5-47 shows the output of this report in Excel. Notice the total for Audio in cell F9. If you look at the Excel input line, you see that this is represented as a formula as opposed to an actual value of \$162,704,821. This is a helpful feature for users who want to download reports and then analyze the data and possibly delete or move rows of data. Having formulas instead of values in your report allows the total fields to adjust automatically. If you want to see the constant values downloaded, then simply choose Excel 97 or Excel 2000 for your output.

Product Type	Rank	Product Category	Revenue	Cost of Goods Sold	Gross Profit
Audio	1	Speakers	84,717,053.00	24,680,990.00	\$60,036,063
	2	Audio Systems	122,345,680.00	82,282,820.00	\$40,062,860
	3	MP3	43,491,588.00	26,438,660.00	\$17,052,928
	4	ers	42,374,428.00	25,739,570.00	\$16,634,858
	5	Recorders	53,847,459.00	37,838,460.00	\$16,008,999
	6	Receivers	35,907,113.00	22,998,000.00	\$12,909,113
Subtotal for Audio			382,683,321.00	219,978,500.00	\$162,704,821
Camcorders	1	DVD Camcorders	379,376,637.00	300,373,350.00	\$79,003,287
	2	MiniDV Camcorders	51,539,451.00	34,128,360.00	\$17,411,091

Figure 5-47 Report Assistant Excel output

Tip: If Gross Profit is calculated as a *compute* field instead of a *define* field, it is also represented in the Excel spreadsheet as a formula.

5.3.4 DB2 database file output

One final output option that we investigate is the creation of a DB2 database file on the System i machine:

Tip: You must have imported the metadata or created a synonym for your output table before you can use it as input to another query.

1. Click the **Report options** tab.
2. On the Report options tab, select **Show database formats** in the second drop-down list (Figure 5-48 on page 97). This enters in DB2 - DB2 database table in the first pane. It also shows a section in the bottom right corner of the window where you can specify the library and file or table name you want to create.

Figure 5-48 Report Assistant output to a DB2 database file

This time when you run your report, the records are inserted in the table that you specify. After the file insert is complete, you see a new window that tells you the number of records that were inserted into the DB2 table (Figure 5-49).

Note: See “Using output files for the next query” on page 303 for recommendations about converting Query/400 definitions that currently create temporary output files in QTEMP.

No HTML Output!			
0	NUMBER OF RECORDS IN TABLE=	15	LINES= 15
0	HOLDING DB2 FILE...		

Figure 5-49 Report Assistant DB2 output confirmation

- Set your output option back to HTML as shown in Figure 5-43 on page 93 or reload your saved report.

We continue with the last steps of our example.

5.4 Additional report types

In this section, we explore other types of reports, including:

- ▶ Reports that accept an input parameter, which are called *parameterized reports*
- ▶ Reports with drill-down capabilities

5.4.1 Parameterized reports

We are asked to print our RA4_Advanced ranking report for a single product type that is requested at runtime. To do this, we add an input parameter:

1. If you are still in your RA4_Advanced report, click the **Selection criteria** tab (Figure 5-50).
If you are not, open the RA4_Advanced report. On the DB2 Web Query home page, right-click the **RA4_Advanced** report and select **Open**.
2. From the Available fields pane, drag the **PRODUCTTYPE** field onto the empty white pane on the right.
3. Click **Select values** to define the parameter.

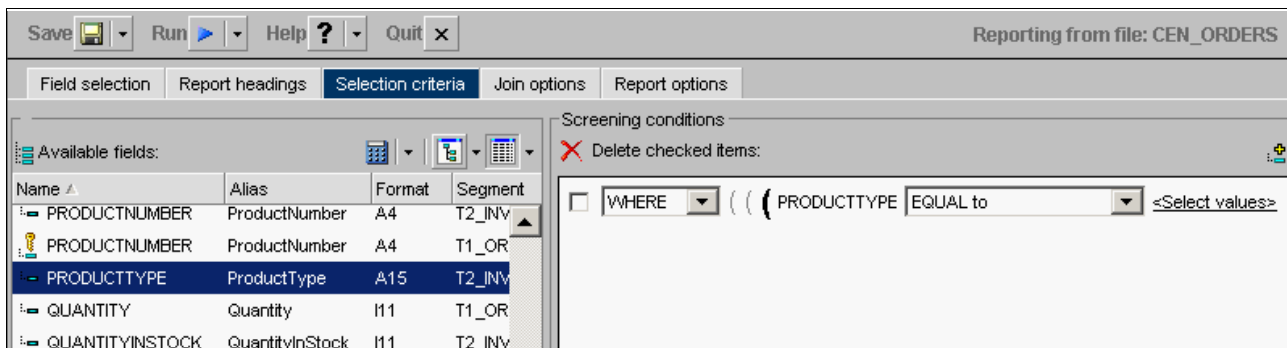


Figure 5-50 Report Assistant Selection criteria

- In the EQ Values window (Figure 5-51), select **Parameter** and then click the **Auto Prompt** icon (🔑).

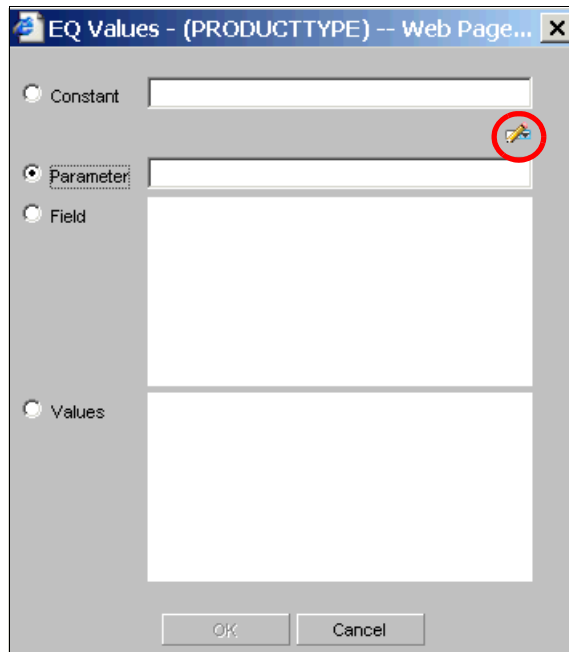


Figure 5-51 Report Assistant parameter definition

- Under Selection, ensure that **Dynamic** is selected (Figure 5-52).

Note: Choosing Dynamic presents the users with a distinct list of current values of the column when the query is run. The Static option allows you to define a predefined set of values.

If you allow the users to select multiple values at runtime, they will be able to use the Windows standard Ctrl and Shift keys to select multiple product types. For now, leave it to default to a single input value.

For the Description field in the Name section, enter the prompt that you want the users to see. We use the default name for this parameter of PRODUCTTYPE.

Figure 5-52 Report Assistant Parameter properties

6. To make the drill-down report more understandable, place the product type in the report heading:
 - a. Click the **Report headings** tab (Figure 5-53).
 - b. Add the words “for Type” to the heading.
 - c. Select the **Show - Field list** arrow and drag **PRODUCTTYPE** to the end of the heading.

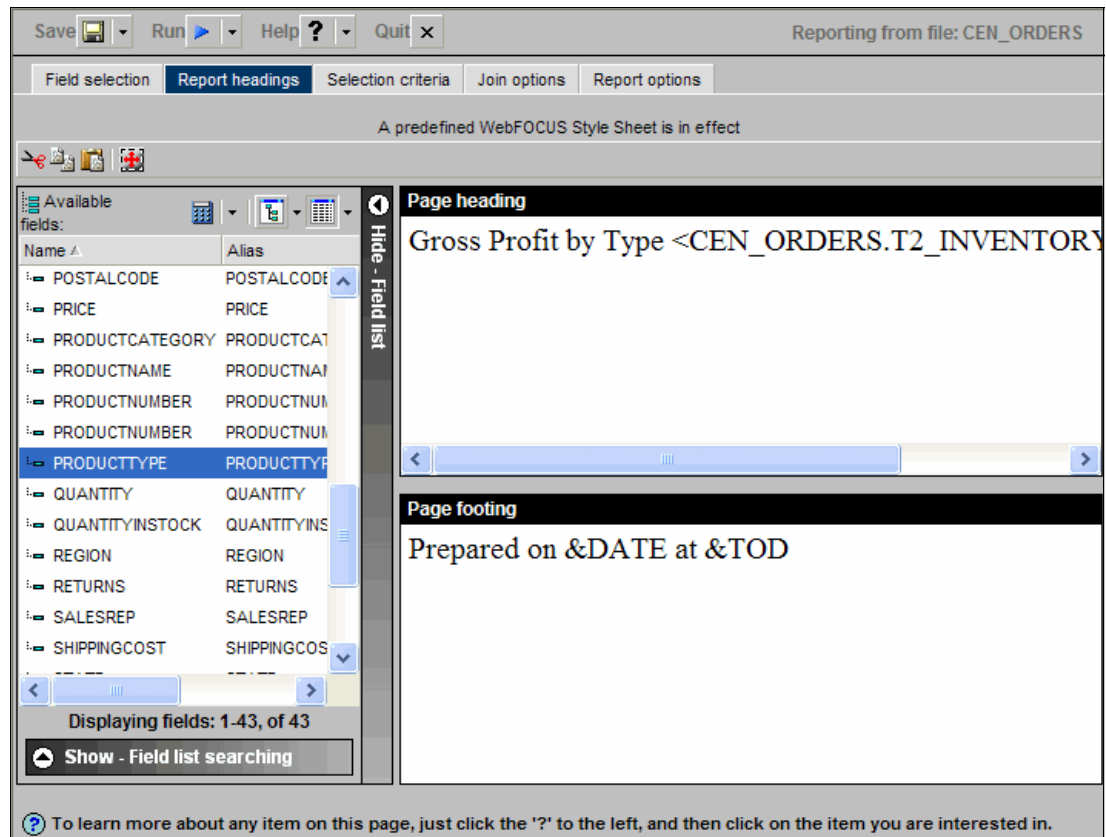


Figure 5-53 Report Assistant variable report heading

- Since we know that this parameterized report is small and, in most cases, it fits on one browser page, remove the default page numbering. Click the **Report options** tab and for Page numbering, select **Off** (Figure 5-54).

Reporting from file: CEN_ORDERS

Field selection | Report headings | Selection criteria | Join options | **Report options**

Generate report as: **HTML - Web browser** | Show styled formats | ☐ Send to printer

Output format specific

Report title:

☐ Format HTML Output | Full

☒ Optimize output with Cascading Style Sheets

☐ Use accessibility features (Section 508)

Summary text for report:

Interactive options

Enable OLAP: | Automatic drill down:

☐ On-demand paging

☐ Show pop-up field description

Global report styling

Font... | Active Report Styling

Apply an existing WebFOCUS Style Sheet: **DEFAULT**

Apply an existing Cascading Style Sheet: **NONE**

Content and generation

Page numbering: **Default** (dropdown menu open showing Default, On, **Off**, NOLEAD)

Limit the number of read operations during report generation:

Stop retrieving records after the specified number have been included in the report:

☐ Use database optimization

Database Output

Destination/Library:

Name: **HOLD**

☐ Overwrite existing file

Printer

Name:

To learn more about any item on this page, just click the '?' to the left, and then click on the item you are interested in.

Figure 5-54 Default page numbering

- Run the report. Figure 5-55 shows the results of running this report.

Producttype: **Audio**

☐ Run in a new window

Gross Profit by Type Audio					
Product Type	Gross Profit Rank	Product Category	Revenue	Cost of Goods Sold	Gross Profit
Audio	1	Speakers	84,717,053.00	24,680,990.00	\$60,036,063
	2	Audio Systems	122,345,680.00	82,282,820.00	\$40,062,860
	3	MP3	43,491,588.00	26,438,660.00	\$17,052,928
	4	Amplifiers/PreAmps/Tuners	42,374,428.00	25,739,570.00	\$16,634,858
	5	CD Players and Recorders	53,847,459.00	37,838,460.00	\$16,008,999
	6	Receivers	35,907,113.00	22,998,000.00	\$12,909,113
Subtotal for Audio			382,683,321.00	219,978,500.00	\$162,704,821

Figure 5-55 Report Assistant Output parameter report

9. Save and close this report.

We now have a report that can either accept a parameter passed by another report or prompt the user for their input.

5.4.2 Drill-down reports: Parent report (RA5_Child)

In this section, we take our initial RA1_Revenue report and enable it to pass an individual product type to the report that we just created.

1. On the DB2 Web Query home page, right-click the **RA1_Revenue** report and select **Open**.
2. In the Sort by pane, highlight **PRODUCTTYPE**.
3. Click the **Show - Field options** arrow.
4. On the Display tab, click the **Drill down** button (Figure 5-56).

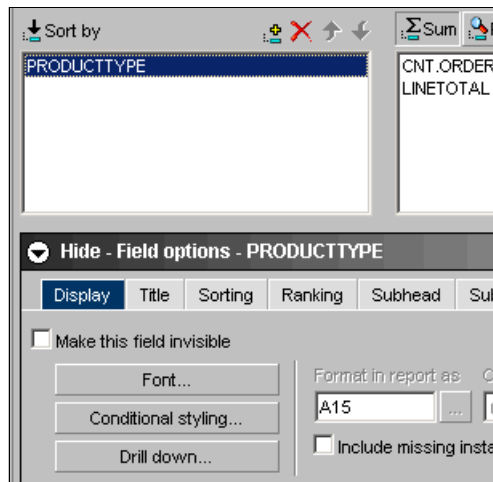


Figure 5-56 Report Assistant Drill-down option

5. In the next window (Figure 5-57), complete these steps:
 - a. Click the **Execute procedure** radio button. DB2 Web Query populates the list with your reports.
 - b. Navigate through the list until you find your **RA4_Advanced** report and select it.
 - c. In the Alternate comment for hyperlink field, add a comment for users to see when they hover over the Product Type field.
 - d. Click **Add**.

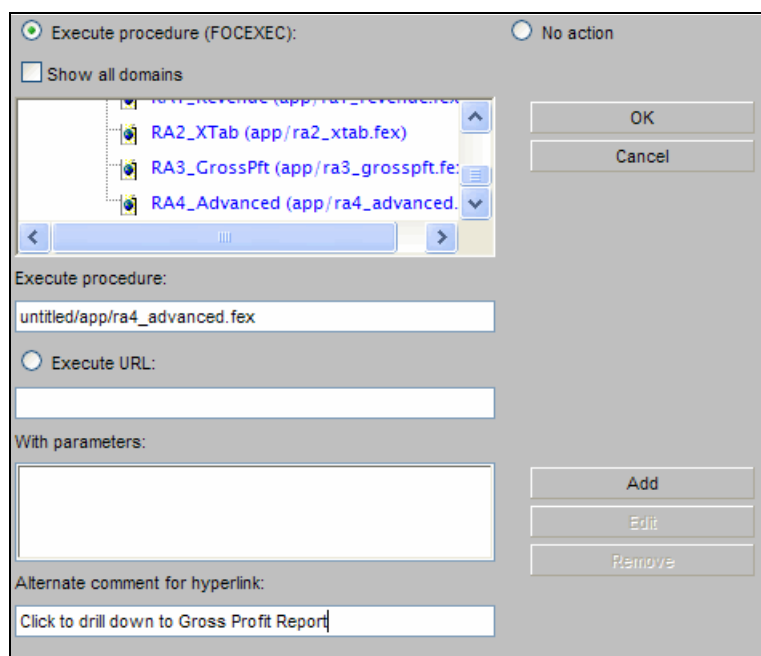


Figure 5-57 Report Assistant Drill-down window

- e. In the Drill-down parameter window (Figure 5-58), complete these steps:
 - i. In the Parameter name field, type the same name as the parameter that the drilldown report is expecting. In this case, type **PRODUCTTYPE**.
 - ii. Under Parameter value, select the Field option and then select the field that you want passed into the variable **PRODUCTTYPE**. Select the **PRODUCTTYPE** field.
 - iii. Click **OK**.

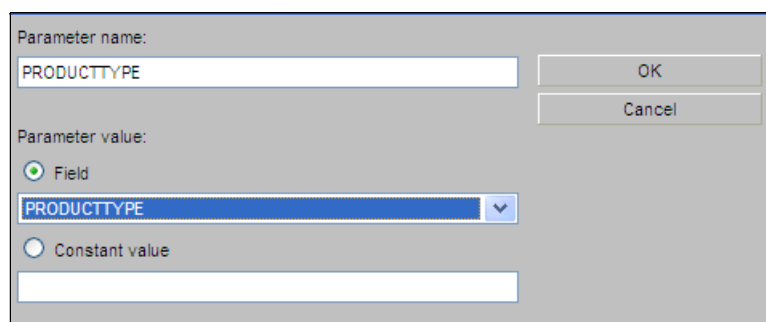


Figure 5-58 Report Assistant drill-down parameter window

6. Run your report. Figure 5-59 shows the results of running the report.
7. In the parent or top-level report, the product types are highlighted in blue indicating that they can be selected and you can then drill-down to a more detail report. Click **Audio** to drill down to see more information.

Product Type	Order Count	Revenue
Audio	12,846	\$382,683,321.00
Car Audio	5,139	\$444,531,041.00
Car Stereo		\$67.00

Click to drill down to Gross Profit Report

Gross Profit by Type Audio					
Product Type	Gross Profit Rank	Product Category	Revenue	Cost of Goods Sold	
Audio	1	Speakers	84,717,053.00	24,680,990.00	\$6
	2	Audio Systems	122,345,680.00	82,282,820.00	\$4
	3	MP3	43,491,588.00	26,438,660.00	\$1
	4	Amplifiers/PreAmps/Tuners	42,374,428.00	25,739,570.00	\$1
	5	CD Players and Recorders	53,847,459.00	37,838,460.00	\$1
	6	Receivers	35,907,113.00	22,998,000.00	\$1
Subtotal for Audio			382,683,321.00	219,978,500.00	\$16
TOTAL			382,683,321.00	219,978,500.00	\$162

Figure 5-59 Report Assistant parent (summary) and child (drill-down) reports

8. Save your report:
 - a. Click the **Save** button at the top of the window and select **Save As**.
 - b. In the Save Standard Report window, enter the name it RA5_Child and click **OK**.
9. Close the report and close Report Assistant.

5.4.3 Report properties

To view the properties of a report, on the DB2 Web Query home page, navigate to the **RA4_Advanced** report, right-click, and select **Properties**. The results are shown in Figure 5-60.

The General tab of the Properties window shows the creation date, the domain, and most importantly the Href or file name on disk. The Detail tab show all the tables and columns that are used in your query. You also see your sorting details, filtering criteria, output format, and any joins that you might have explicitly defined.

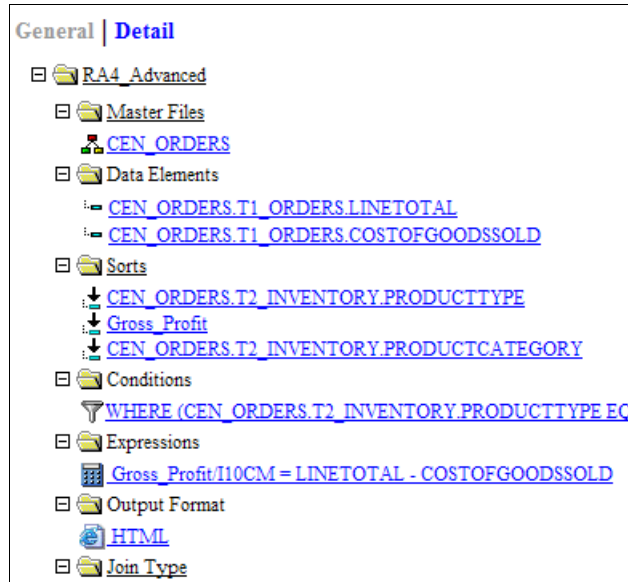


Figure 5-60 Report Assistant report properties

You have now completed this tutorial.

5.5 Joining tables (reference only)

You might have noticed that we never needed to join any tables together even though we looked at data from multiple tables. As previously discussed, this is because the tables were defined to DB2 with foreign key constraints.

If either of the tables do not have foreign keys defined or you choose not to include the foreign keys when you created the metadata, you must create your joins manually. In this section, we explain the process that you must follow.

After you select your main table to query, you click the Join options tab to specify any additional tables that you want to access. You must click New to create a new join. In the window that opens, you highlight the table in the list to which you want to join and click OK. See Figure 5-61.

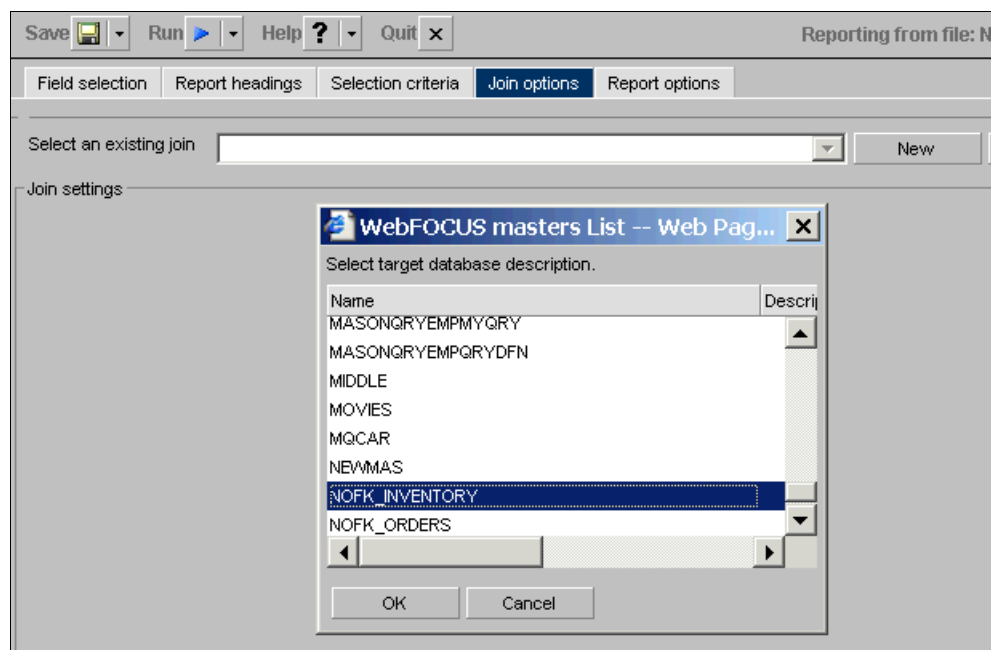


Figure 5-61 Report Assistant join options

Next you must define the actual join criteria. You tell DB2 Web Query which field from each table that you want to use for your join. See Figure 5-62.

Join types:

- ▶ *Unspecified:* This option was created for compatibility with prior versions. It is recommended that, instead of this choice, you select the appropriate options (Inner Join, Left Outer Join, Multiple Instances, Single Instances).
- ▶ *Inner Join:* An inner join combines rows from both tables where there is an exact match between the join fields.
- ▶ *Left Outer Join:* This option retrieves records from both primary and secondary tables, including all records from the left table (primary) and any records from the right table (secondary) where the condition values match. If there are no matching values in the secondary table, the join still retrieves the records from the host table.
- ▶ *Multiple Instance:* A one-to-many join structure that matches one value in the host data source to multiple values in the cross-referenced field. Joining employee ID in a company's employee data source to employee ID in a data source that lists all the training classes offered by that company would result in a listing of all courses taken by each employee, or a joining of the one instance of each ID in the host file to the multiple instances of that ID in the cross-referenced file.
- ▶ *Single Instance:* A one-to-one join structure that matches one value in the host data source to one value in the cross-referenced data source. Joining an employee ID in an employee data source to an employee ID in a revenue data source is an example of a unique join.

You must remember to click the Save & Create button. If you forget to do this and navigate away from this page, your join is not created.

The screenshot displays the 'Join options' tab in the DB2 Web Query Report Assistant. At the top, there are tabs for 'Field selection', 'Report headings', 'Selection criteria', 'Join options' (active), and 'Report options'. Below these, a dropdown menu 'Select an existing join' is set to 'J001', with 'New' and 'Delete' buttons to its right. The 'Join settings' section contains:

- 'Join Type' dropdown set to 'Unspecified'.
- 'Instances' dropdown set to 'Multiple'.
- 'Description' text field containing 'J001'.
- 'Internal name' text field containing 'J001'.
- 'Save & Create' and 'View...' buttons on the right.

 Two field lists are presented:

- 'Host field list (NOFK_ORDERS)': A table with columns 'Name', 'Alias', and 'Form'. It lists fields like COSTOFGOODSSOLD, LINETOTAL, ORDERDATE, ORDERNUMBER, PLANTCODE, **PRODUCTNUMBER** (highlighted), and QUANTITY. Below it, a 'Selected Fields' section shows 'PRODUCTNUMBER'.
- 'Target field list (NOFK_INVENTO...)': A similar table listing fields like COST, MODEL, PRICE, PRODUCTCATEGORY, **PRODUCTNUMBER** (highlighted), and PRODUCTTYPE. Its 'Selected Fields' section also shows 'PRODUCTNUMBER'.

 Navigation arrows and icons are present between the field lists and below the selected fields.

Figure 5-62 Report Assistant join definition

After you save and create the first join definition, you can click New and proceed to define a second join if you want.

When you click the Field selection tab, you see that the fields from both tables are available for use (Figure 5-63).

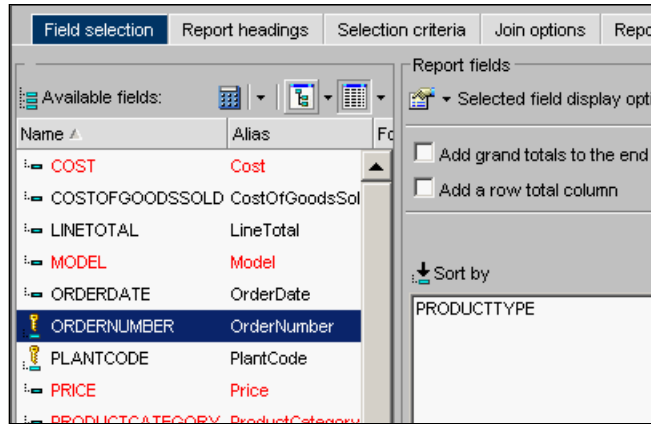


Figure 5-63 Report Assistant secondary table field

Notice that the columns from the secondary table are in a different color than the columns from the primary table. This is a quick visual check to help identify the different source tables. If you select the Tree icon (between the calculator and the spreadsheet), you can see the fields listed in physical sequence by table.

Tip: If you switch to tree mode and hover over a field name, additional information pertaining to that field is displayed.

5.6 Report Assistant summary

You have now completed the first tutorial and created a few different reports. In many of the remaining chapters, you modify the reports that you created in this chapter. We do not repeat information that is similar between the different components, in some cases, such as joining tables. In other examples, we list the steps but in far less detail. If you forget how to accomplish a function, return to this chapter for a refresher of the steps.



Graph Assistant

Graph Assistant provides a wide variety of graph types and graphing functionality to allow users to choose how to present their data in the best possible format. The available graph types range from a variety of bar and line charts to histograms and area charts, stock charts, gauges for key performance indicators, and much more. You can find additional details about the graphing facility in the help text.

In this chapter, we continue with the same Century database used in the previous tutorials. In this chapter, we complete the following tasks:

- ▶ Create a simple bar chart that shows revenue by product type within country. We drill down from this graph to the detail report that we created previously with Report Assistant. We then modify this graph to be a dual axis bar and line chart that accepts country as an input parameter.
- ▶ Create a line chart that shows revenue trending over time. The users can specify the time range they want to see.
- ▶ Create multiple pie charts with traffic lighting or conditional styling based on a calculated field.

6.1 Tutorial overview: Bar chart

In this exercise, we initially create a bar chart that allows the user to drill down to a previous report (Figure 6-1).

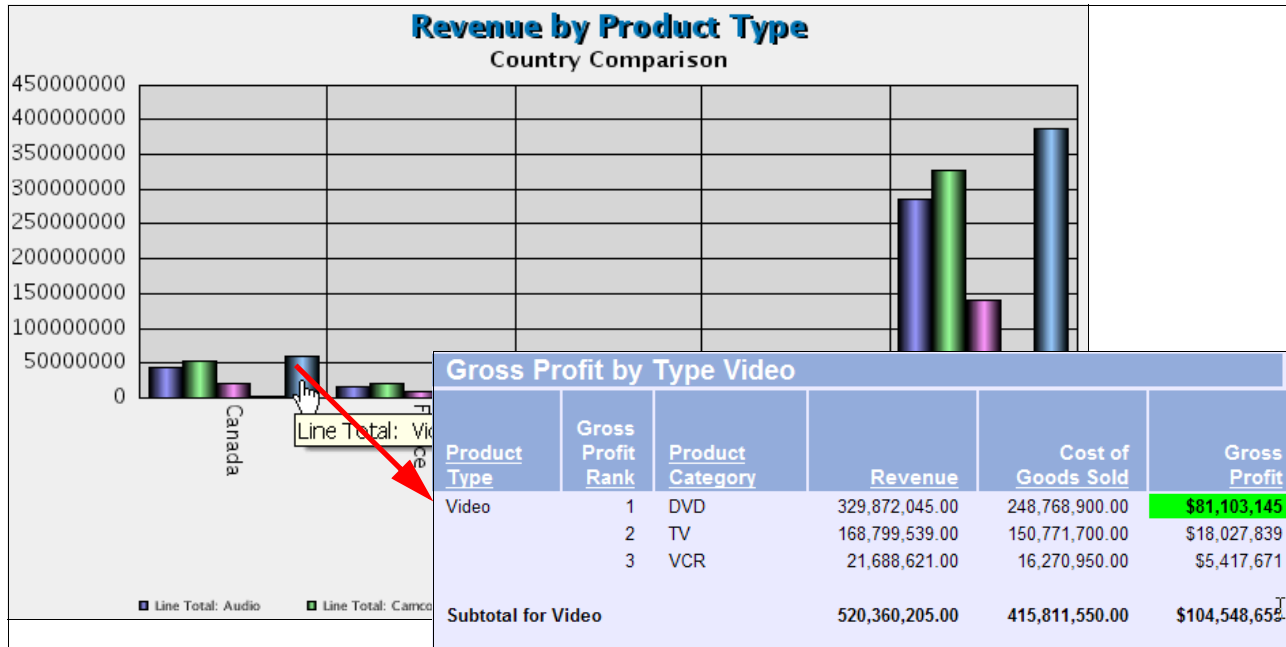


Figure 6-1 Completed bar chart with drill down

We then modify this bar chart to create a dual axis bar and line chart that accepts country as an input parameter. See Figure 6-2.



Figure 6-2 Modified chart by adding a gross profit line

6.2 Creating a simple bar chart with drill-down capabilities (GA1_Bar)

To create a simple bar chart with drill-down capabilities:

1. Right-click the **Tutorials** folder and select **Graph Assistant**.
2. In the window (Figure 6-3) that opens, select the same **CEN_ORDERS** table that we used in the previous tutorials. Click **OK**.

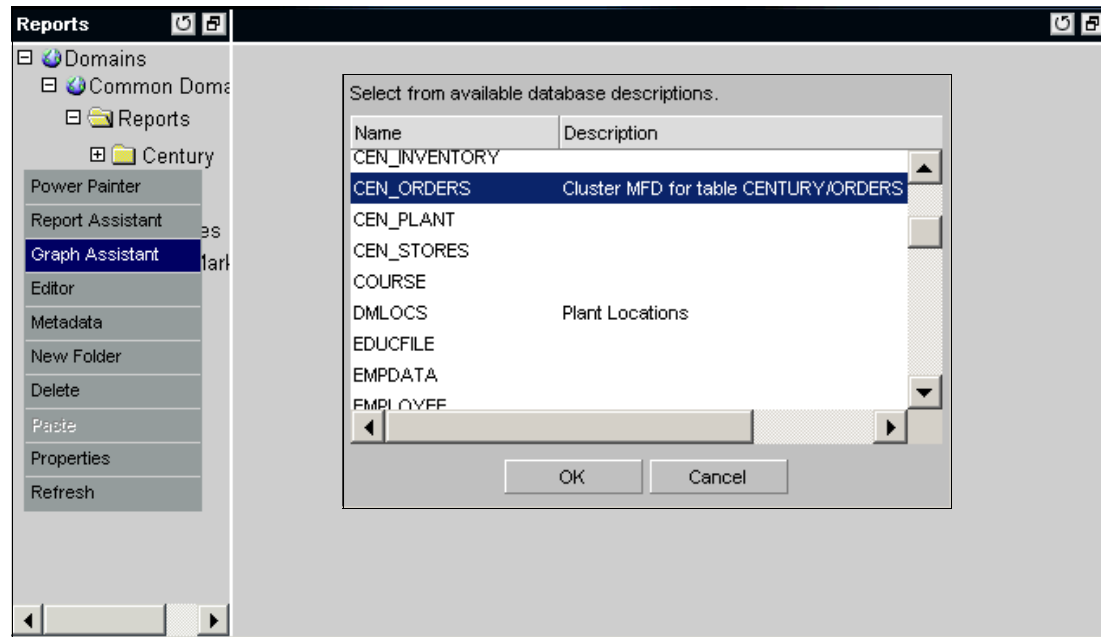


Figure 6-3 Starting Graph Assistant

3. In the Graph Assistant, click the **Graph types** tab (Figure 6-4). On this tab, complete the following steps:
 - a. Under Select from the following graph types, look at the various chart types that are available. For graph type, choose a **Bar** chart type.
 - b. For each chart type, there are many variations shown in the Graph Styles pane. For Graph Styles, select **Clustered**.
 - c. Look at the various templates under Apply template. Select **Cylinder on grey**.

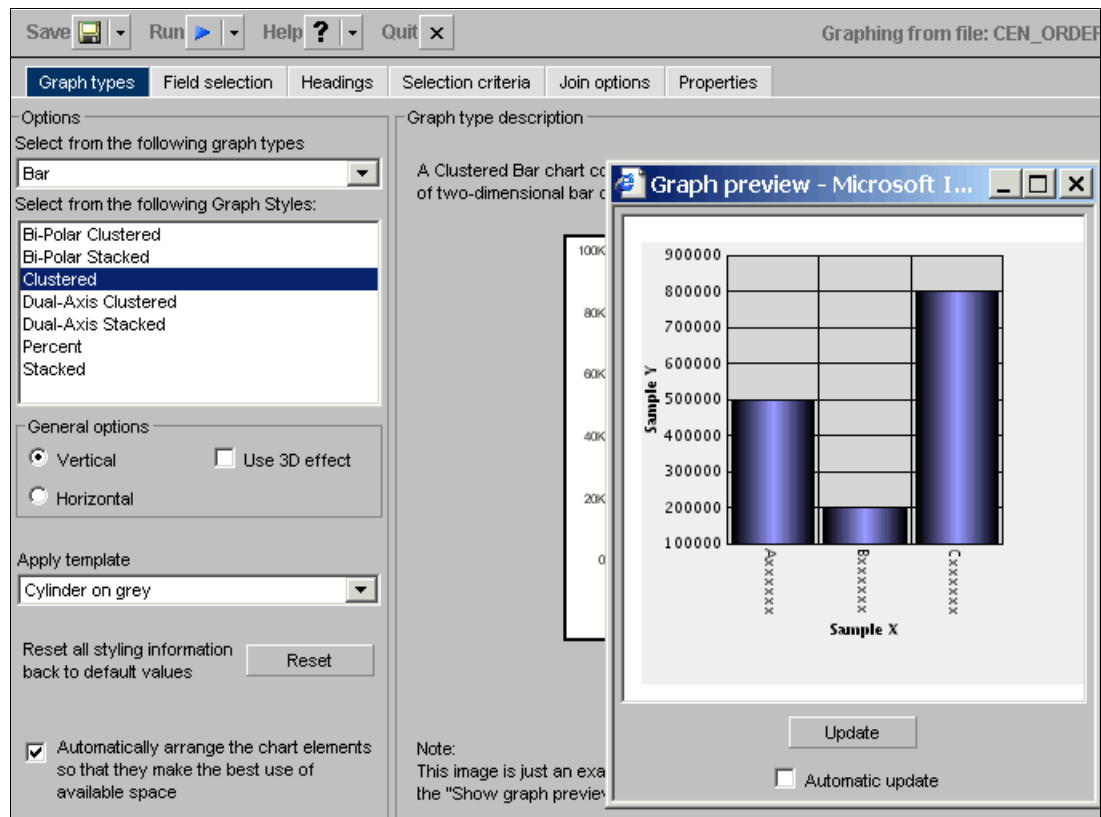


Figure 6-4 Graph Assistant Graph type selection

4. Click the **Field selection** tab (Figure 6-5).
5. Define the fields that represent the x and y axis for the chart:
 - a. Drag **LINETOTAL** to the Sum pane. This is our Y axis.
 - b. Drag **COUNTRY** to the X axis field.
 - c. Drag **PRODUCTTYPE** into the separate graph area field. By default, this produces one chart for each separate plant.

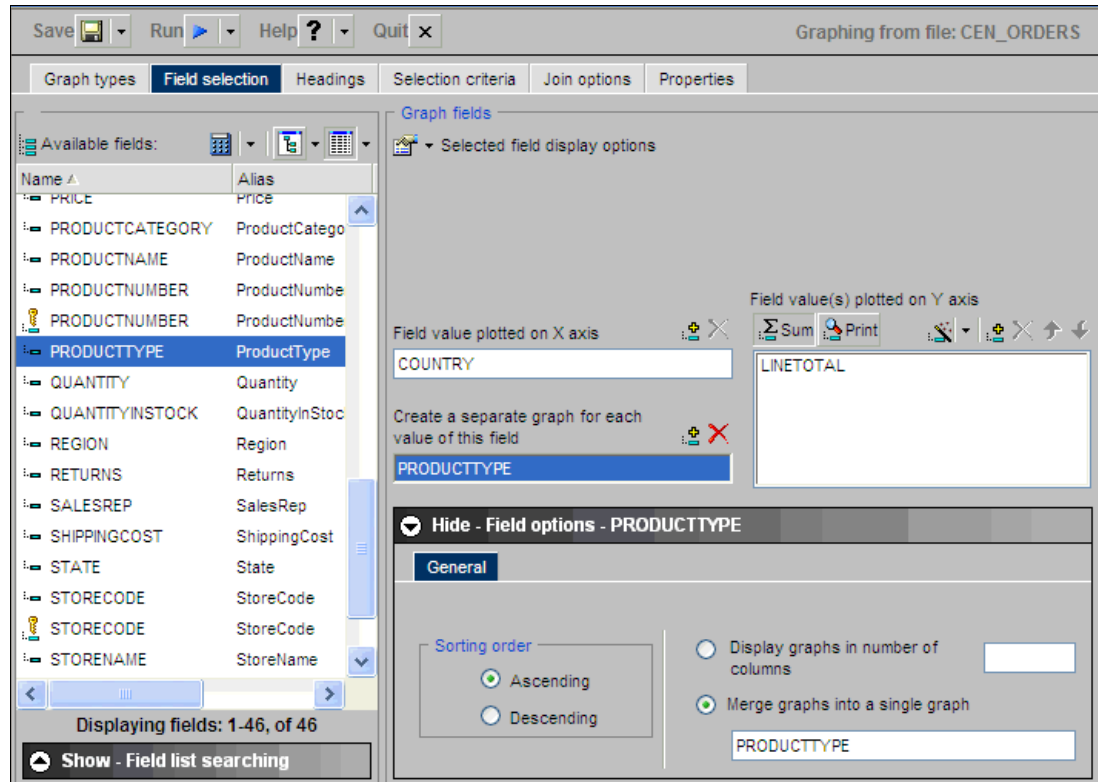


Figure 6-5 Graph Assistant Field selection

6.2.1 Adding multiple bars to one chart

To add multiple bars to one chart:

6. On the Field selection tab, under the Graph fields section, ensure that **PRODUCTTYPE** is still highlighted.
7. Click the **Show - Field options** for **PRODUCTTYPE**.
8. Select **Merge graphs into a single graph**. Instead of having a graph for each product type, this option merges the graphs so that each type is a separate bar within each country.

- Click the **Headings** tab (Figure 6-6). Graphs can have headings both at the page level and at the individual chart level. Click the **Graph** tab. Under Chart title, type “Revenue by Product Type” and for Chart subtitle, type “Country Comparison”.

The screenshot shows the Graph Assistant interface with the 'Headings' tab selected. The 'Chart title' field contains 'Revenue by Product Type' and the 'Chart subtitle' field contains 'Country Comparison'. There are 'Set style' buttons next to each field. A 'Show - Field list' button is visible on the left side of the interface.

Figure 6-6 Graph Assistant Chart Titles

- Save your graph as GA1_Bar and then run it. Figure 6-7 shows the results of running this report.

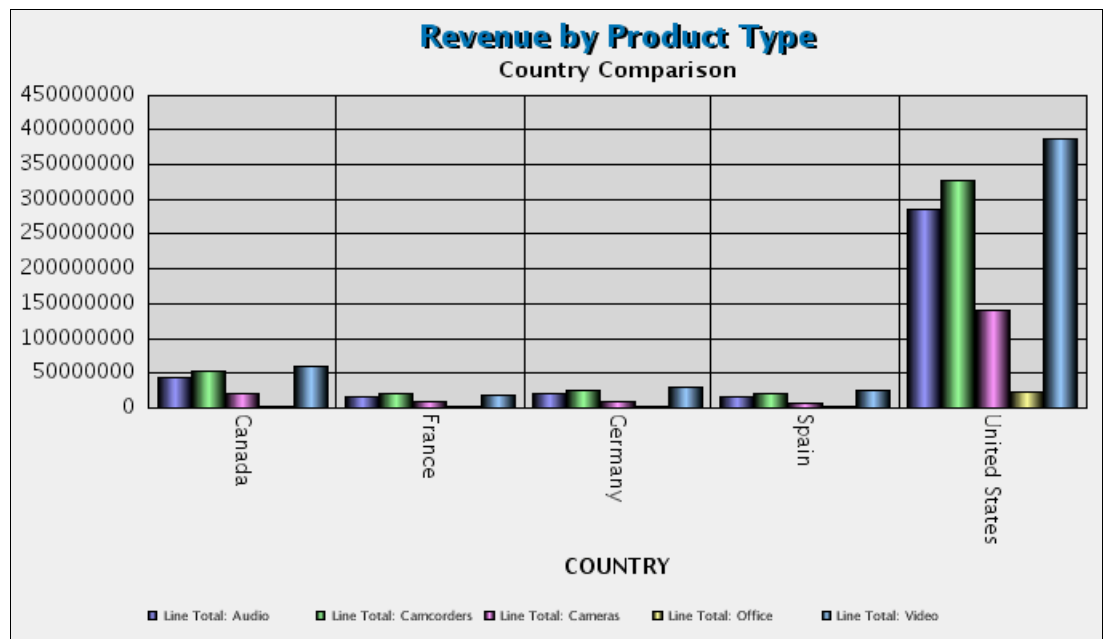


Figure 6-7 Graph Assistant Revenue by Product Type Chart

6.2.2 Adding a drill down to a chart

Now that we have our chart, we want the users to have the ability to double-click any of the bars and drill down to the detailed product type report that we created earlier:

11. Return to the **Field selection** tab, and highlight **LINETOTAL**.

12. Under Field options - LINETOTAL, on the Display tab (Figure 6-8), click the **Drill down** button.

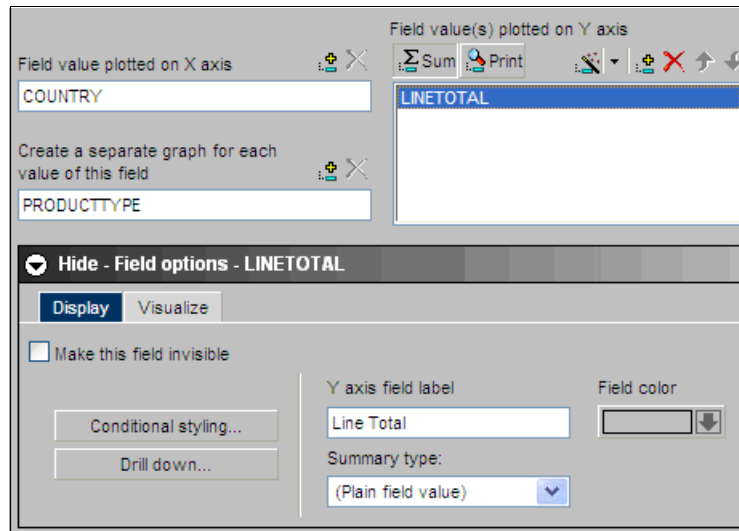


Figure 6-8 Graph Assistant Drill-down button

13. In the next window that opens (Figure 6-9), complete these steps:

- Select **Execute procedure** and a list of your reports is populated in the top white pane.
- Navigate to and select the **RA4_Advanced** report that you created earlier. After you select your procedure, the Add button becomes available.
- Click **Add** to add the parameter that you pass from this report to the RA4_Advanced report.

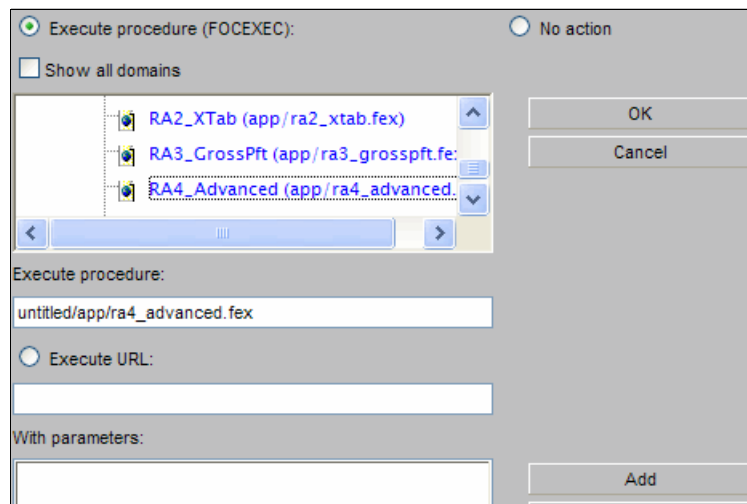


Figure 6-9 Drill-down selections

14. In the next window (Figure 6-10), complete these steps:

- a. For Parameter value, select **Field** and then select the field that you want to pass as the parameter value, which is **PRODUCTTYPE**.
- b. Under Parameter name, type PRODUCTTYPE. This name must match the name that the child report is expecting to be passed. We defaulted our parameter name to equal the input column name of PRODUCTTYPE.
- c. Click **OK**.

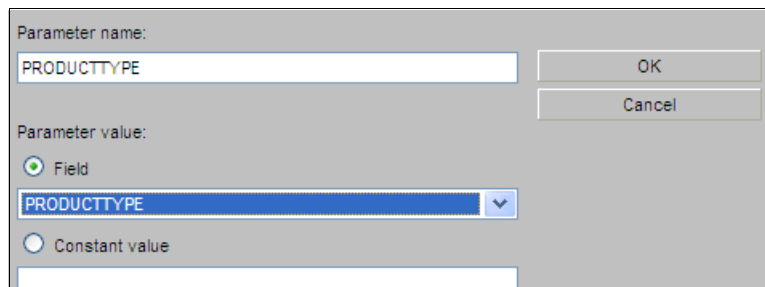
The screenshot shows a dialog box titled "Graph Assistant parameter passing". It has two main sections. The first section is "Parameter name:" with a text input field containing "PRODUCTTYPE" and two buttons, "OK" and "Cancel", to its right. The second section is "Parameter value:" with two radio buttons. The first radio button, labeled "Field", is selected. Below it is a dropdown menu showing "PRODUCTTYPE". The second radio button, labeled "Constant value", is unselected and has an empty text input field below it.

Figure 6-10 Graph Assistant parameter passing

Drill down: When you drill down from one report to another, the child report expects a parameter. In our case, this is the PRODUCTTYPE parameter that is linked in the gross profit report to the PRODUCTTYPE field. In the parent report or graph, you indicate where users can drill down.

In this graph, the users can drill down on any LINECODE or bar. When they click a bar, we tell the system that we want the RA4_Advanced report to run. We also indicate that we are going to pass in the PRODUCTTYPE field that is referenced by the selected bar. We do this by passing the value as a parameter called PRODUCTTYPE. Since this is the name of the parameter that the Advanced report was expecting, it accepts the value and executes the report.

15. Save and then run this graph. Figure 6-11 shows the results of running this report.
16. Roll over the bars with your mouse. Notice that the cursor changes from an arrow to a hand. The hand indicates that you can drill down in this area. When you hover over a bar, a window opens that shows all the details about the bar. First you see the product type, then the country, and lastly the revenue or line total.

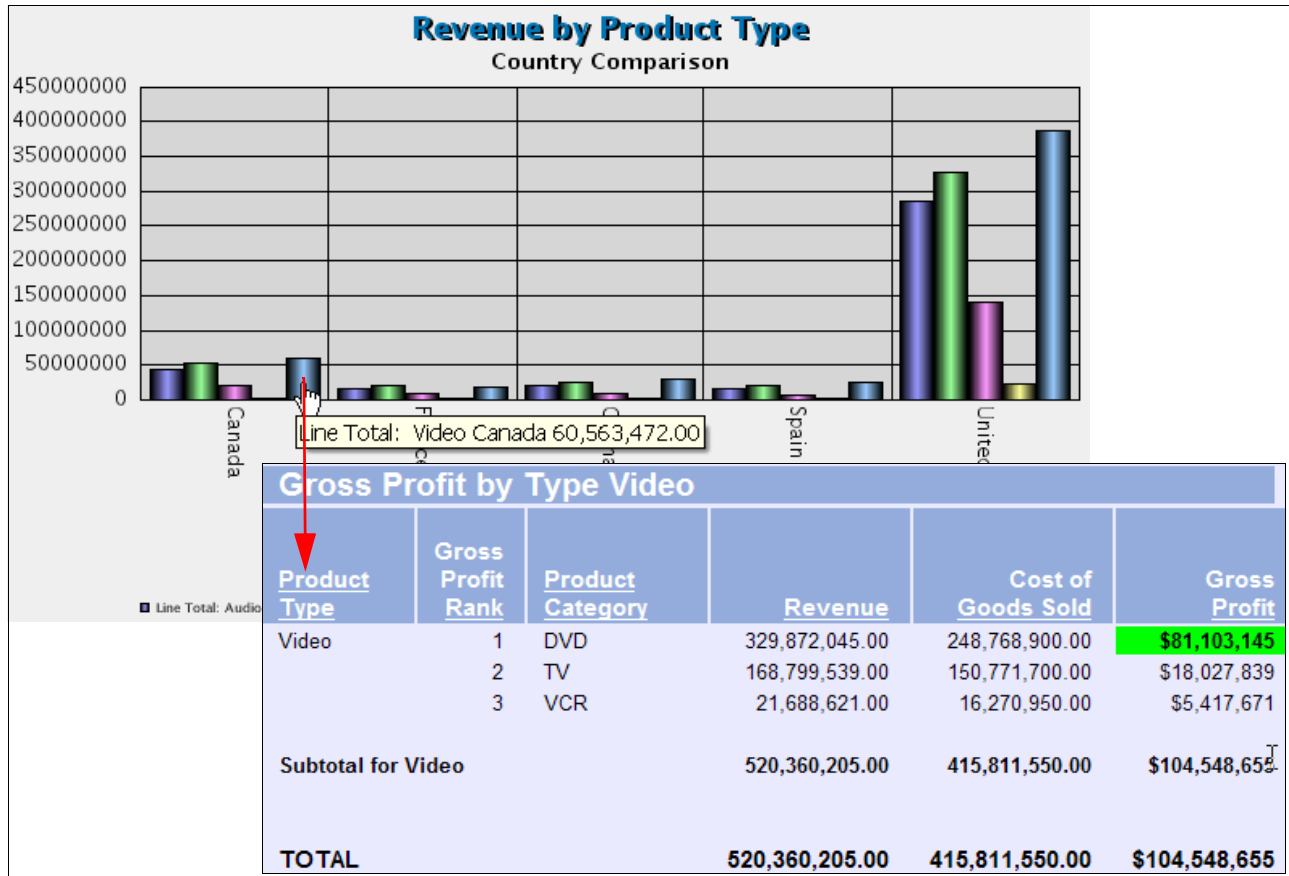


Figure 6-11 Graph Assistant drill down from chart to report

Note: In this drill-down example, we pass the product type to the child report. We could design the report to accept both country and product type. If we do this, when the user clicks the same bar, they can pass the country name as well as the product type as filters or parameters to the Gross Profit report (RA3_GrossPft).

6.3 Creating a dual axis bar and line chart (GA2_Parm)

In this exercise, we change COUNTRY from a set of bars to an input parameter. We also add gross profit as a line chart on top of our existing bar chart. Gross profit has its own Y axis. This way we can see the relationship between our revenue and the profit that each product generated.

1. If you closed the bar chart that you just created, navigate to the **Tutorials** folder, locate **GA1_Bar**, right-click the report name, and click **Open**.
2. Click the **Field selection** tab.

3. Modify the existing graph slightly as shown in Figure 6-12:
 - a. In the Create a separate graph for each value of this field, remove **PRODUCTTYPE** by highlighting the entry and clicking the red Sort Across Remove (X) button.
 - b. Change COUNTRY in the Field value plotted on X axis field, to be PRODUCTTYPE. You can choose one of the following options:
 - Delete COUNTRY by clicking the Sort Across Remove (X) button.
 - Drag **PRODUCTTYPE** on top of COUNTRY, which does an automatic replace.

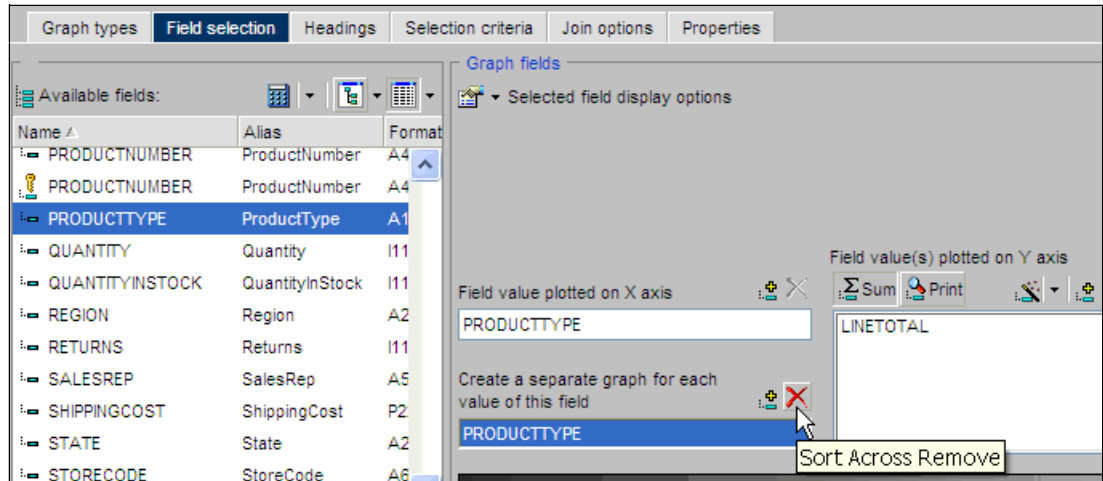


Figure 6-12 Changing X axis columns

Note: At any time during development, you can click the **Run** button to see how your chart looks up to this point. Using the Update button on the graph preview is useful during development. If you closed your graph preview box, click the **Graph Type** tab and select **Show Graph Preview**.

4. Add COUNTRY as a parameter:
 - a. Click the **Selection criteria** tab (Figure 6-13).
 - b. Drag **COUNTRY** from the Available field list to the pane on the right.
 - c. In this case, the defaults of WHERE and EQUAL TO are appropriate since we want the user to select or specify a single country.
 - d. Click **Select values**.

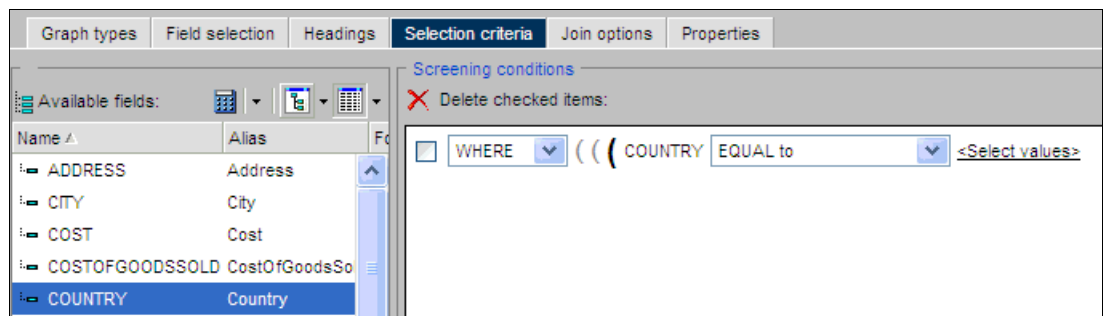


Figure 6-13 Creating a parameter for COUNTRY

- e. In the next window (Figure 6-14), select **Parameter** and click the **Auto Prompt** icon.

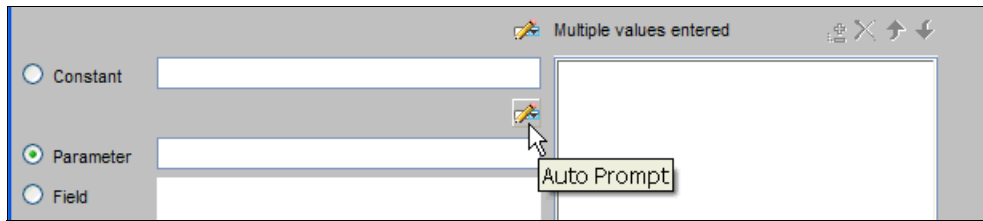


Figure 6-14 Parameter Auto Prompt

- f. In the Parameter Properties window (Figure 6-15), complete these steps:
- i. Change the prompt that the user sees. For Description, type "Select Country:".
 - ii. For the remainder of the window, select the default options.
 - iii. Click **OK**.

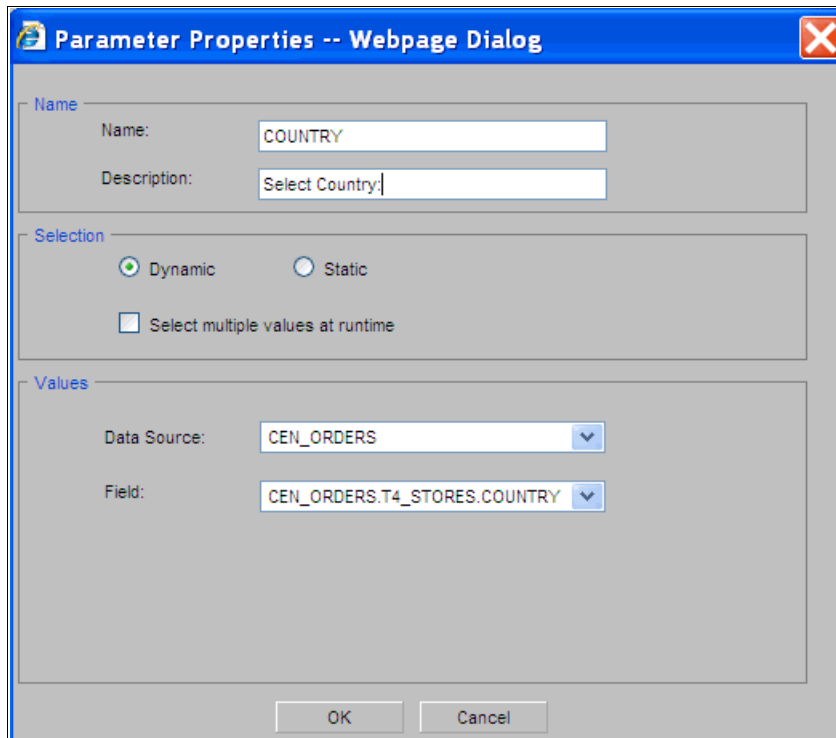


Figure 6-15 Details about COUNTRY parameter

5. Save your report as GA2_Parm.

We just created a simple bar chart that prompts the user for country, which you can try and see how it works. Remember that you can run your report or update the graph preview at any time.

6.3.1 Adding a second Y axis

Redefine the graph as one that supports a second Y axis:

6. Click the **Graph types** tab (Figure 6-16). Change the Graph Styles from Clustered to **Dual-Axis Clustered**.

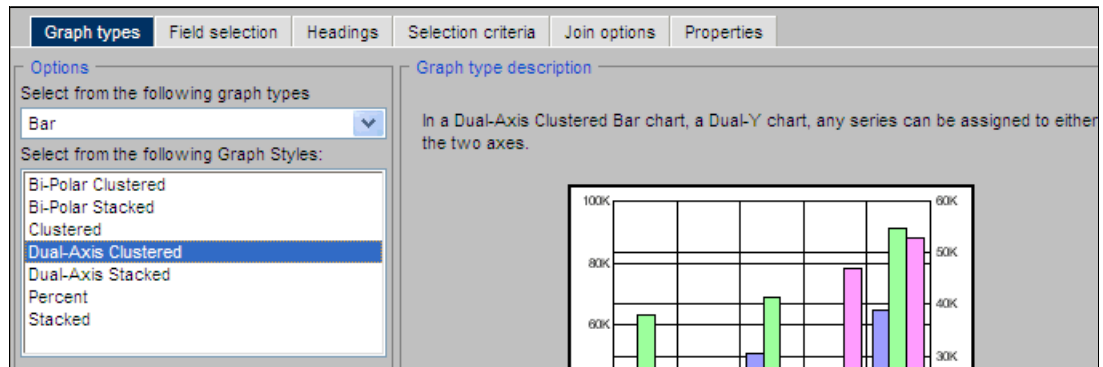


Figure 6-16 Selecting a Dual-Axis Clustered graph type

7. Superimpose Gross Profit as a line on top of the existing bar chart. Since Gross Profit does not exist in the field list, create it:
 - a. Click the **Field selection** tab.
 - b. On the Field selection tab, click the **calculator** icon and select **New define field** (Figure 6-17).

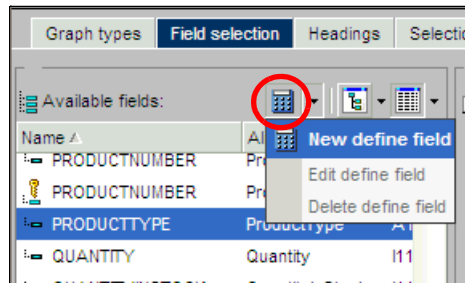


Figure 6-17 Selecting New define field

- c. In the window that opens (Figure 6-18), complete these steps:
 - i. For Field, type Gross_Profit.
 - ii. To define the field, you can type the formula manually or open the field list pane to the right and double-click the field names to move them over. Enter the following formula:

LINETOTAL - COSTOFGOODSSOLD

Now Gross_Profit is available for us to use.

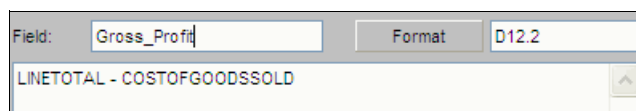


Figure 6-18 Calculating gross profit

8. Drag **Gross_Profit** to the Y axis panel below LINETOTAL.
9. Click **Show - Field options - Gross_Profit**.
10. Click the **Visualize** subtab (Figure 6-19), and complete these steps:
 - a. For Force the values on this field to be plotted as, select **Line**.
 - b. For Place this field on the specified axis, select **2nd axis**.

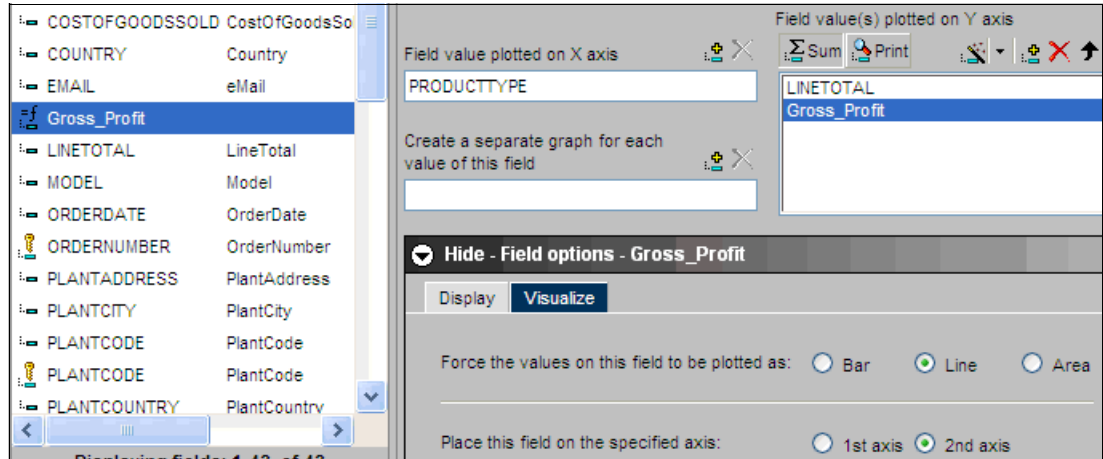


Figure 6-19 Adding a second Y axis for a line chart

11. Ensure that there is a heading for the graph:
 - a. Click the **Headings** tab.
 - b. Click the **Graph** subtab.
 - c. Under Chart title, the graph title is "Revenue by Product Type". We do not need a subtitle for this graph. See Figure 6-20.

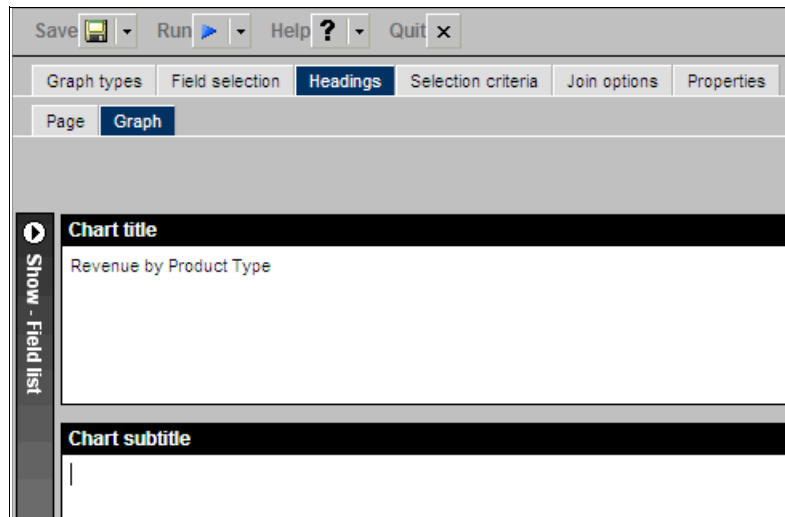


Figure 6-20 Graph headings

12. Save and run your graph.

13. In the report that is displayed, for Select Country, select **Germany**. Then you see the results shown in Figure 6-21.

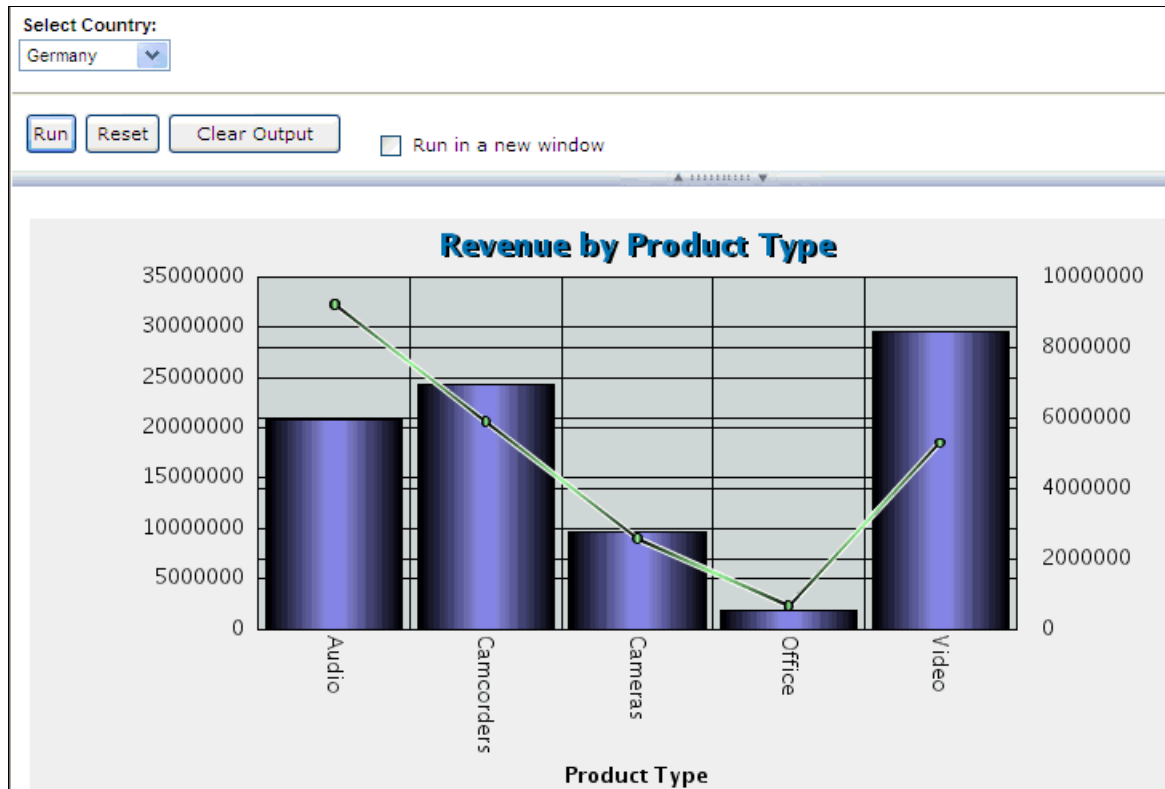


Figure 6-21 Dual axis bar and line chart with prompting

Modifying graph properties and settings

Overall the graph looks like what we wanted. The one potential concern is ensuring that the users can quickly and easily read the values on the Y axis. Let us simplify this for our users:

14. Click the **Properties** tab.

15. On the Properties tab, select the **Y axis** subtab (Figure 6-22).

- Both the revenue and profit are large numbers in the millions. From the drop-down list, select **Y1 axis** to format the product type y axis values.
- For Value format, select **\$10M** to show the Y (gross profit) axis as millions of dollars.

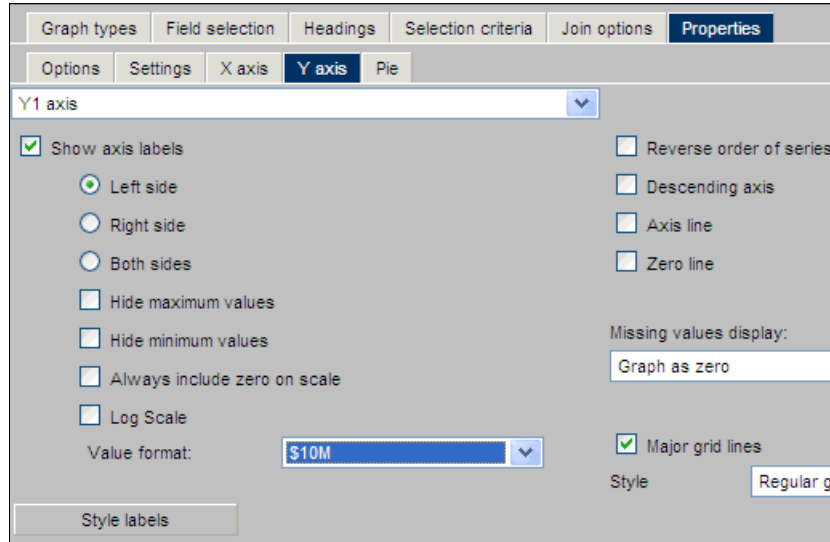


Figure 6-22 Formatting Y axis values

16. Save and run your report. Figure 6-23 shows the result of the chart.

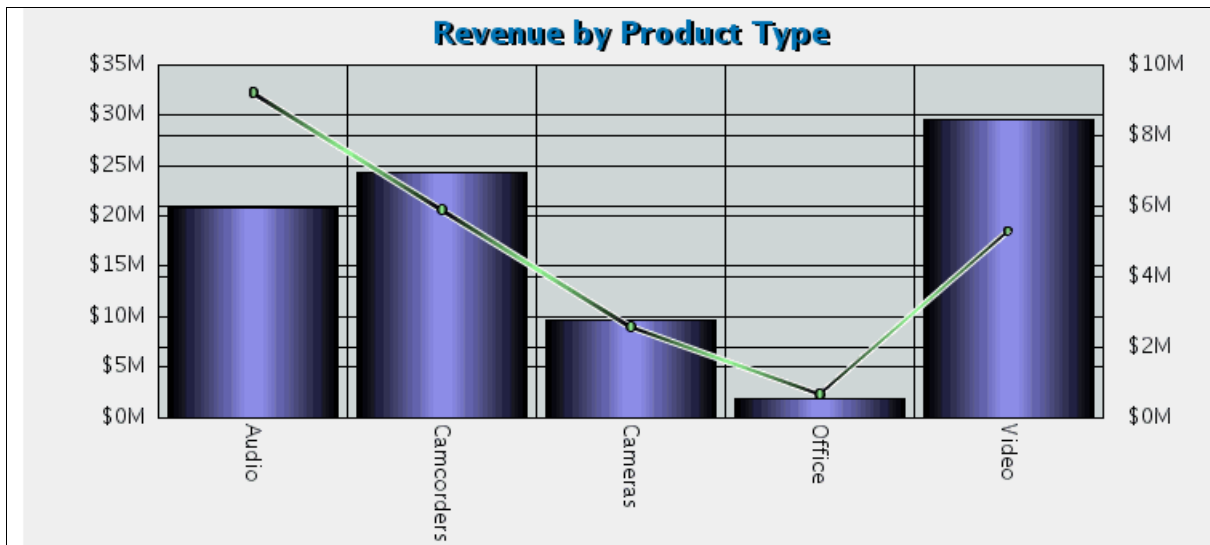


Figure 6-23 Dual axis bar / line chart with prompting

17. Close Graph Assistant.

With these minor changes, our users are now less likely to misinterpret the dollar values.

6.4 Creating a revenue trend over time chart (GA3_Line)

In this exercise, first we create a simple line chart that spans two years (includes all our data). Then we modify this chart to allow the user to specify the time range in which they are interested (see Figure 6-24).

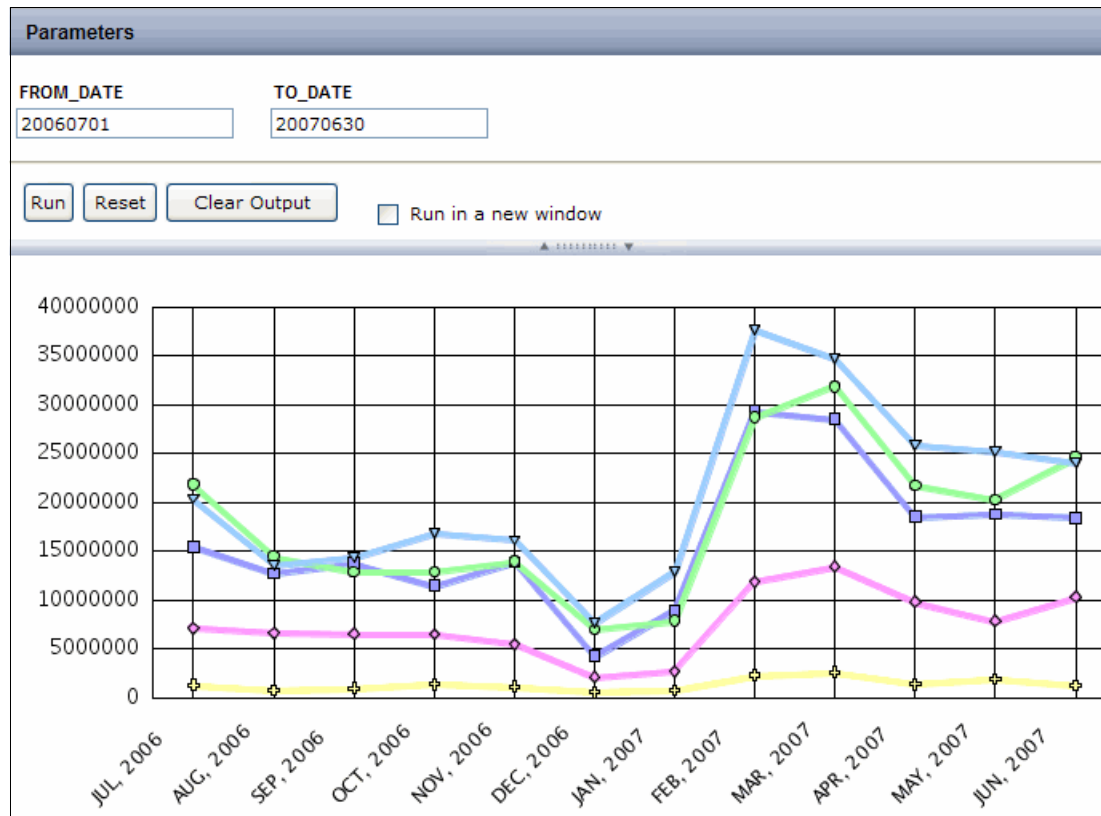


Figure 6-24 Trending report with a variable date range

6.5 Creating a line graph

To create a line graph:

1. Right-click your **Tutorials** folder and select **Graph Assistant** (see Figure 6-25).
2. In the window that opens (inset in Figure 6-25), select the table **Cen_Orders** and click **OK**.

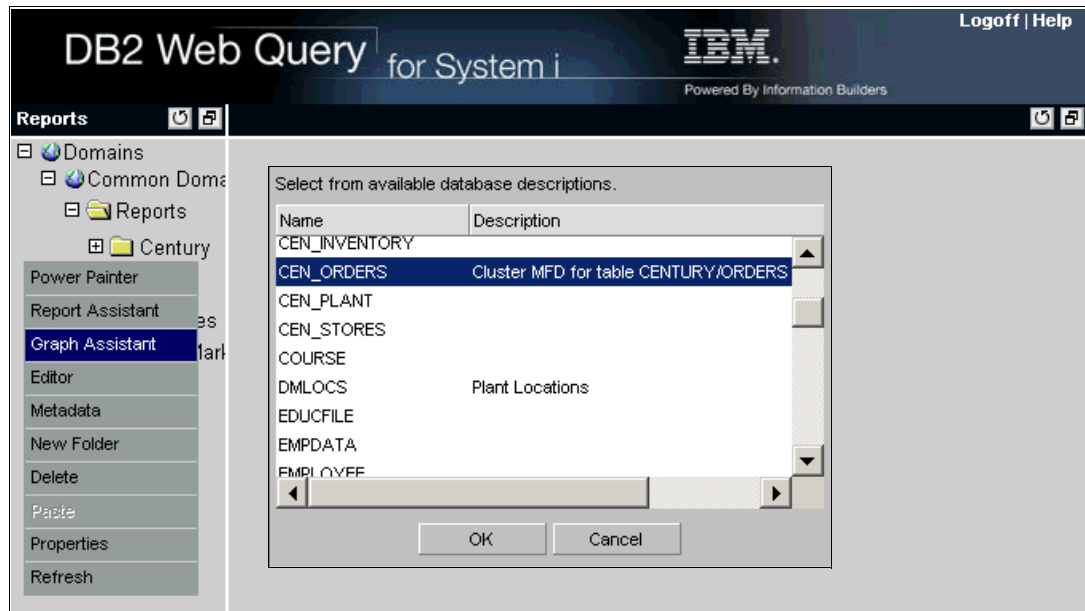


Figure 6-25 Creating a new graph

3. In the next window, click the **Graph types** tab.
4. On the Graph types tab (Figure 6-26), under Select from the following graph types, select **Line**. Then under Select from the following Graph Styles, select **Absolute**.

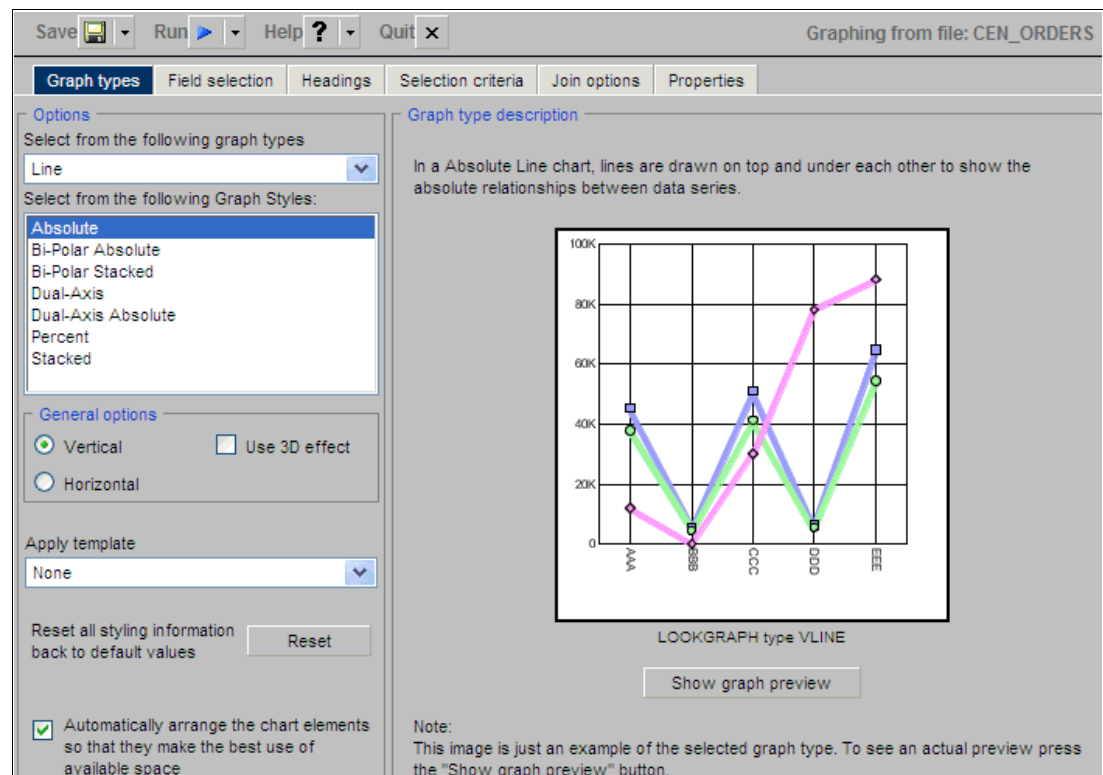


Figure 6-26 Selecting a graph type for a line chart

5. To show product revenue trending by month, you must plot one point for every month for each product. To do this, convert the date field from year, month, and day to a field with just the year and month.
 - a. Click the **Field selection** tab.
 - b. Click the **calculator** icon and select **New define field** (Figure 6-27).

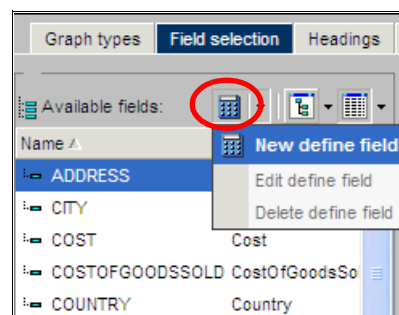


Figure 6-27 Defining a new field

- c. In the Define field creator window (Figure 6-28), complete these steps:
 - i. For Field, type MthYr.
 - ii. In the pane below Field, where you define the formula for the new field, type ORDERDATE, which is the name of the date on which MthYr is based.
 - iii. In the Format field, type TMY.

T means that the month or day immediately following will be represented as uppercase text. A single *Y* means that you want to display a two-digit year, but *YY* means that you want to display a four-digit year.

An example of a MthYr value is Jan, 2007. If we want to place the year first, we specify a format of YYTM. DB2 Web Query is powerful when it comes to working with date and time fields. See “Date and time system variables” on page 368 for many of the different ways you can work with date and time fields.
 - iv. Click **OK**.

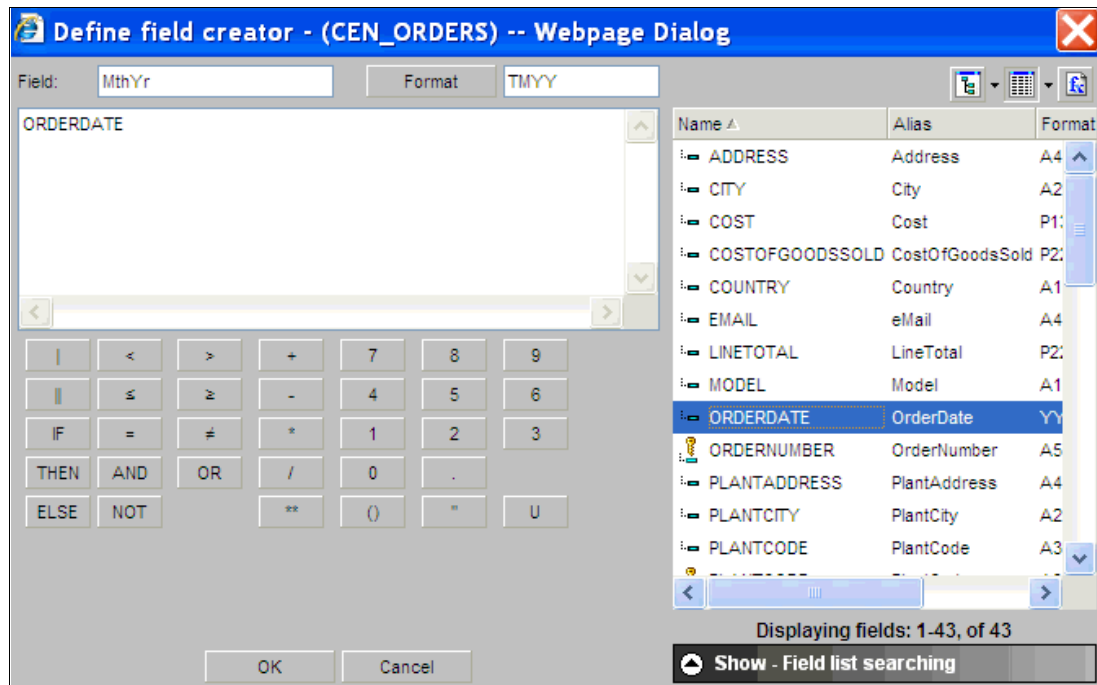


Figure 6-28 Creating a date field of month and year

- d. On the Field selection tab (Figure 6-29), complete these steps:
- Drag **MthYr** to the Field value plotted on X axis (our main X axis) pane. Drag **LINETOTAL** to the Sum/Print pane under Field value(s) plotted on Y axis. Under Create a separate graph for each value of this field, specify **PRODUCTTYPE**.

Tip: You must place LINETOTAL in the SUM/Print pane before you can drag PRODUCTTYPE to the Create a separate graph for each value of this field pane.

- Click **Show - Field options - PRODUCTTYPE**. Select the **Merge graphs into a single graph** option.

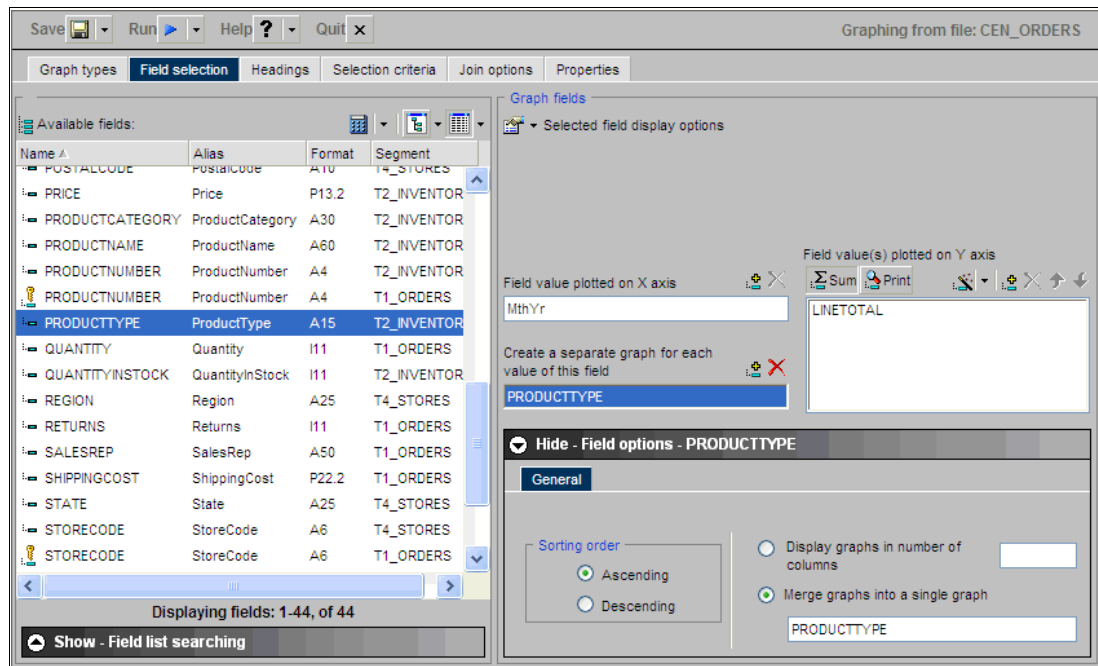


Figure 6-29 Creating a line graph with multiple lines

6. Since an entry on an axis of “Jan, 2007” is obviously a date, remove the heading or label for MthYr from the graph (see Figure 6-30):
 - a. Under Field value plotted on X axis, highlight **MthYr**.
 - b. Click **Show - Field options - MthYr** and under X axis field label, delete the label **MthYr**.

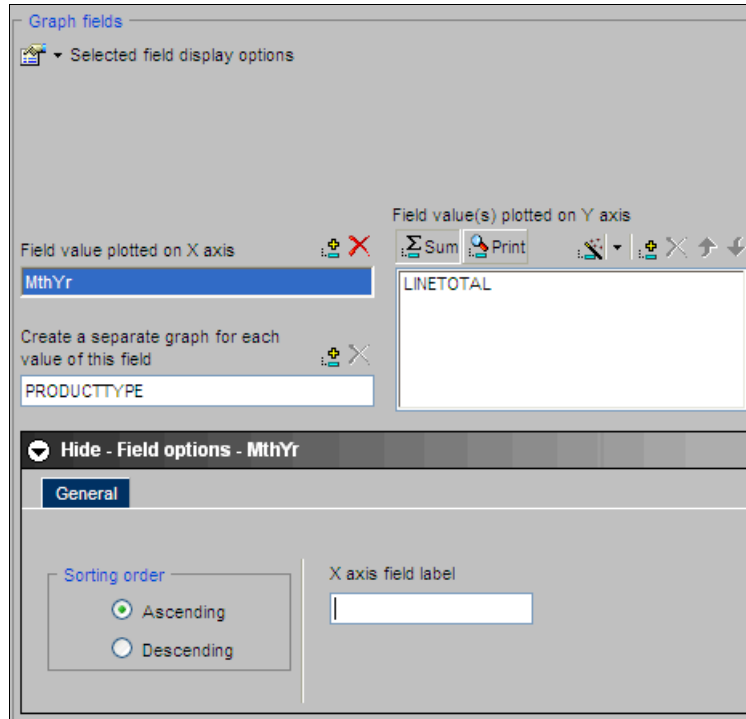


Figure 6-30 Removing the X axis field label

7. Save your report as GA3_Trend.

8. Run your report. Figure 6-31 shows the results of running this report.

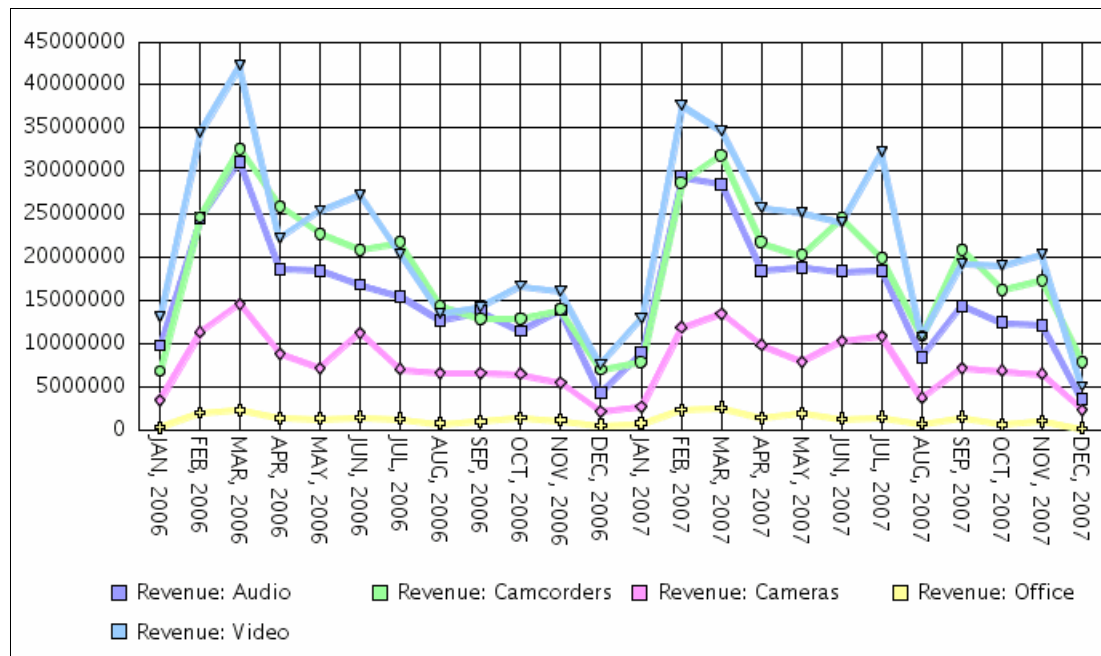


Figure 6-31 Two-year product revenue trend lines

9. Change the X axis to rotate the text and make it look better:
 - a. Click the **Properties** tab and then click the **X axis** subtab.
 - b. Click the **Style labels** button.

- c. In the X Axis Properties window (Figure 6-32), complete these steps:
 - i. Under Set Style for, select **Data**.
 - ii. Some of the options initially are unavailable because a default style sheet is in effect. Select the **Because automatic arrangement is turned on, certain style options are disabled. Check here to override defaults** check box.
 - iii. For Text rotation, select **45-degree bottom to top**.
 - iv. Click **OK**.

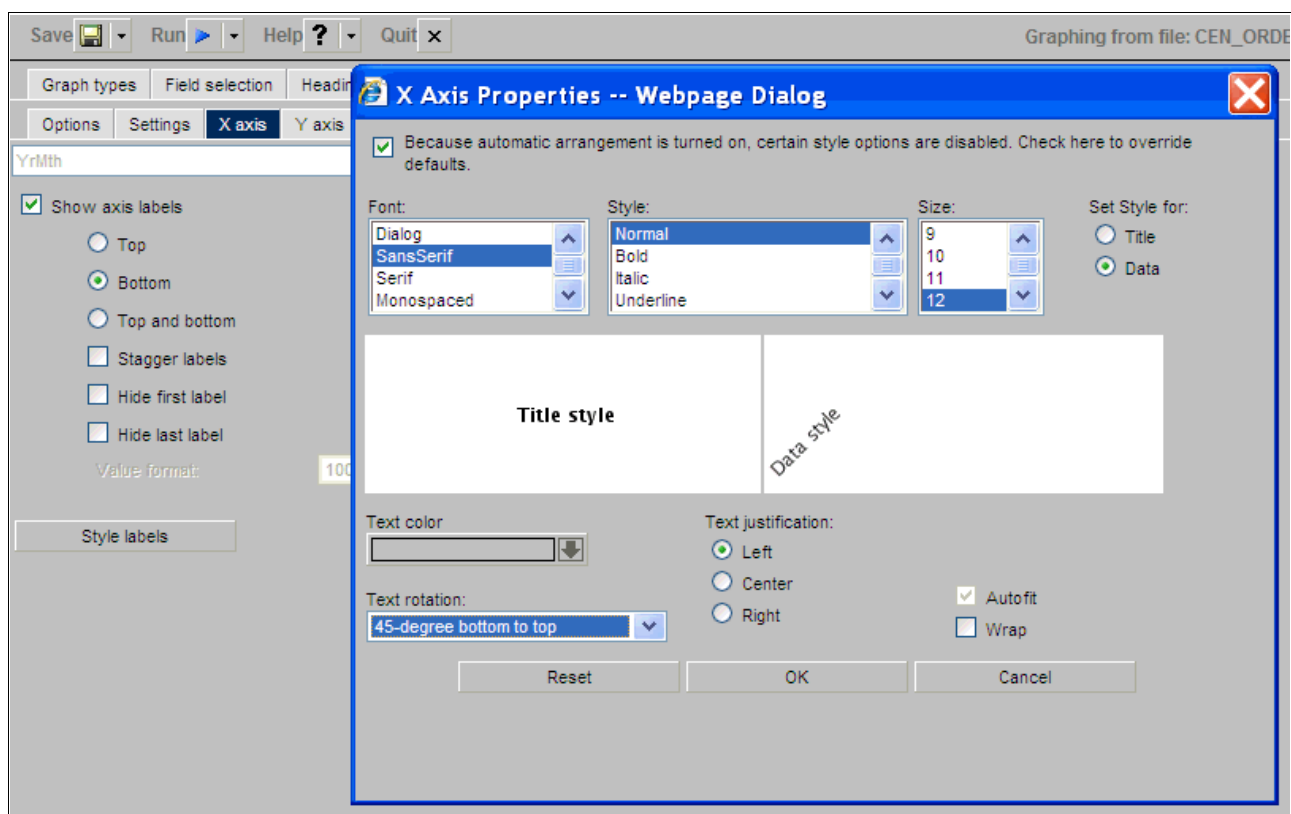


Figure 6-32 45-degree x axis labels

10. Save and run your report. Figure 6-33 shows the result of running this report.

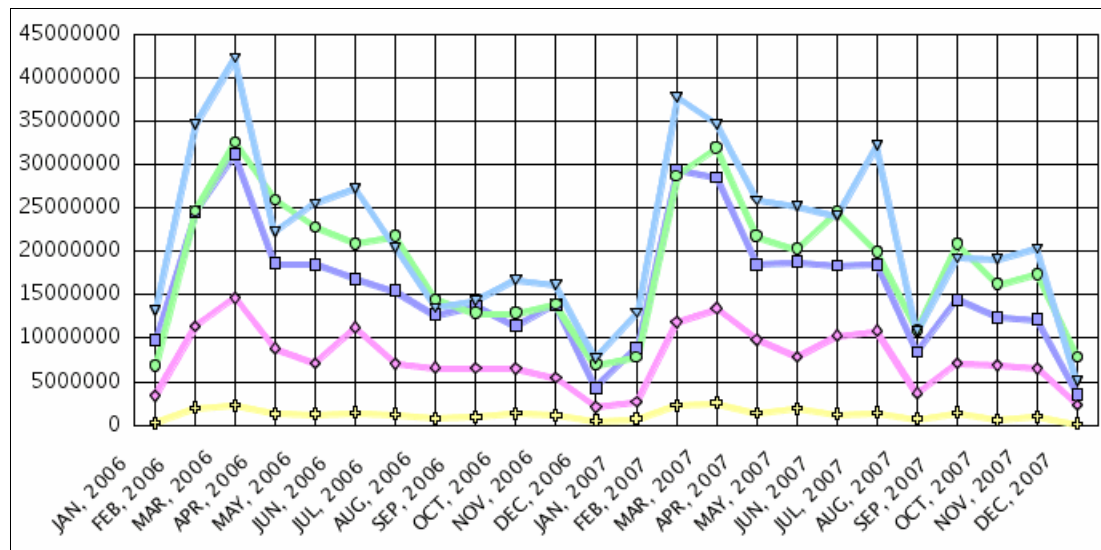


Figure 6-33 Report with 45-degree X axis labels

6.5.1 Adding a date range filter to a line graph

Up to this point, we plotted one point for every month, year combination in our data. This is not practical because the history grows in size. Next we select a specific range to display. We display one year's worth of data starting with orders that were generated on July 1, 2006 and ending with orders received on June 30, 2007. After we look at the options for hardcoding a range, we look at how the user can specify an ad hoc range.

To filter the dates:

11. Click the **Selection criteria** tab (Figure 6-34).

12. Drag **MthYr** to the Screening conditions pane on the left.

13. In the Screening conditions pane, complete these steps:

- a. To select the records as we read them, use the **WHERE** default as opposed to WHERETOTAL.
- b. As stated previously, our range is all orders inclusively from July 2006 to June 2007. For our first WHERE clause, for MthYr, select **GREATER THAN or EQUAL to** and then click **Select values** to enter the date.
- c. In the GE Values window, select **Constant** and type 072006.

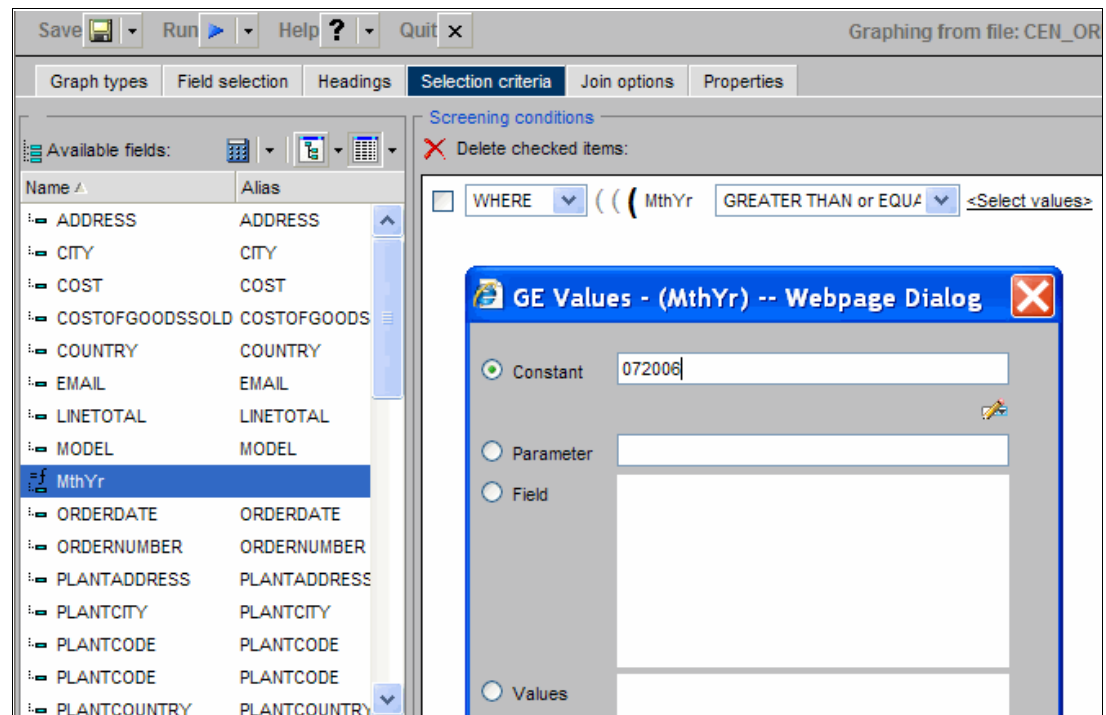


Figure 6-34 Selection criteria for date

14. Drag **MthYr** to the Screening conditions pane (see Figure 6-35).
15. Notice that, by default, the two conditions that you define will be ANDed together.
16. In the Screening conditions pane, complete these steps:
 - a. Select **LESS THAN or EQUAL to** and click **Select values**.
 - b. In the LE Values window, select **Constant** and type 062007.
 You have now defined a range where 072006 <= MthYr <= 062007.

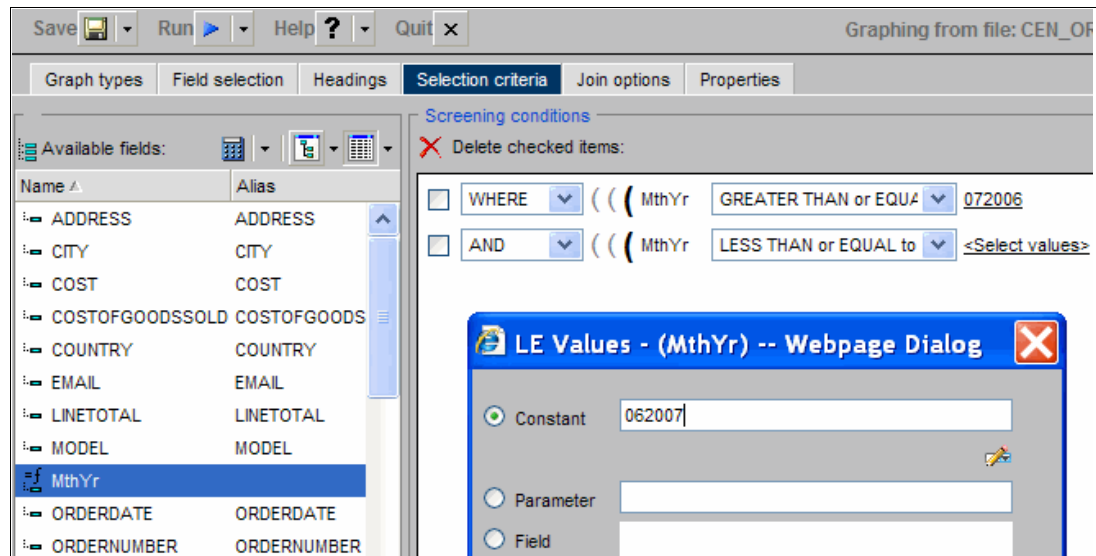


Figure 6-35 Selecting date ranges

17. Save and run your report. Figure 6-36 shows the results of running this report. Notice that our report now starts with July and ends with June.

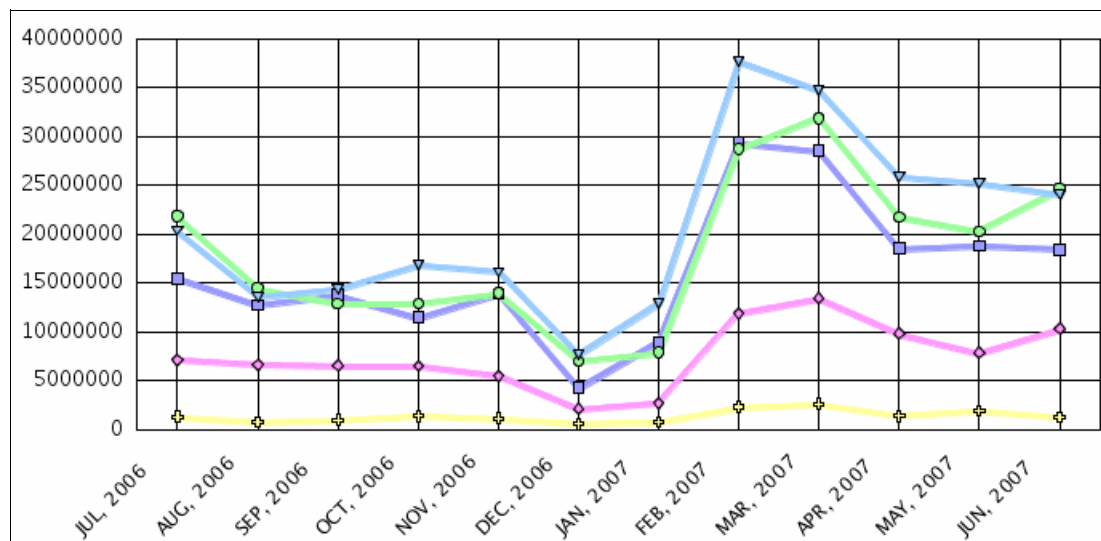


Figure 6-36 Specific date range

Here are a couple of options that achieve the same results:

- Option 1
 - i. Remove the selection criteria that we specified. In the Screening conditions pane (Figure 6-37), select both conditions. Then click the red **X** under the text “Screening conditions” to remove them.

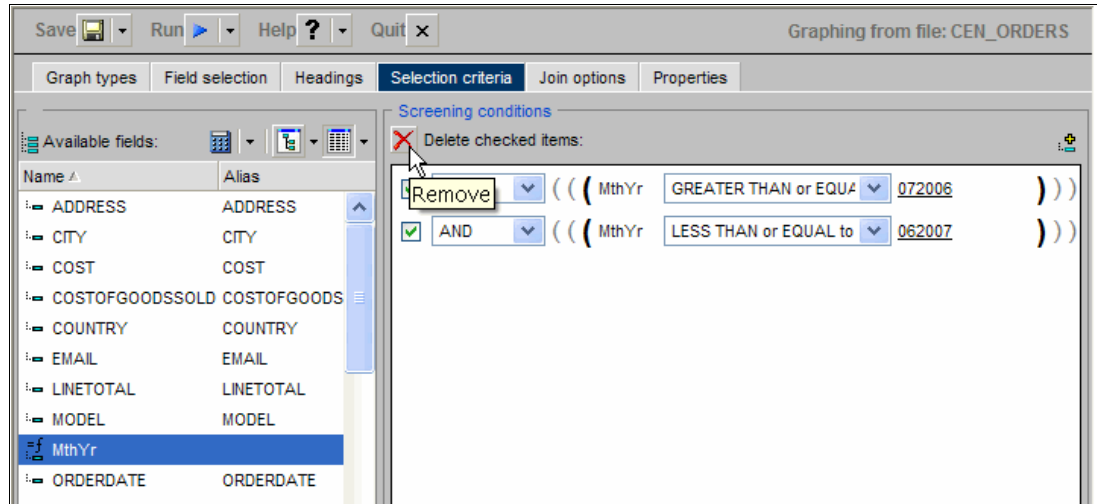


Figure 6-37 Removing screening conditions

- ii. Using the same steps as we did with MthYr, this time drag **ORDERDATE** to the Screening conditions pane and specify the full date. See Figure 6-38.

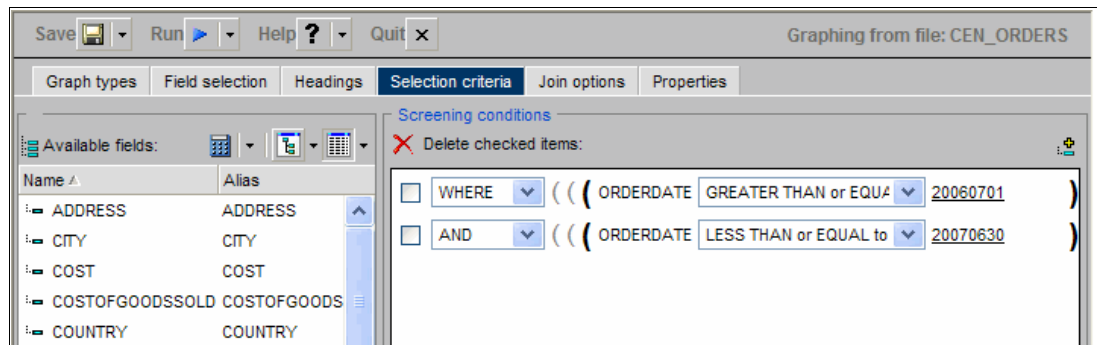


Figure 6-38 Using ORDERDATE column

- iii. Run your report. It should look exactly the same as our previous report.

- Option 2
 - i. Remove the previous screening conditions.
 - ii. Drag **ORDERDATE** to the Screening conditions pane.
 - iii. Select **FROM - TO** and click **Select values**.
 - iv. In the FROM-TO values window (Figure 6-39), for the From criteria, type 20060701, and for the To criteria, type 20070630.

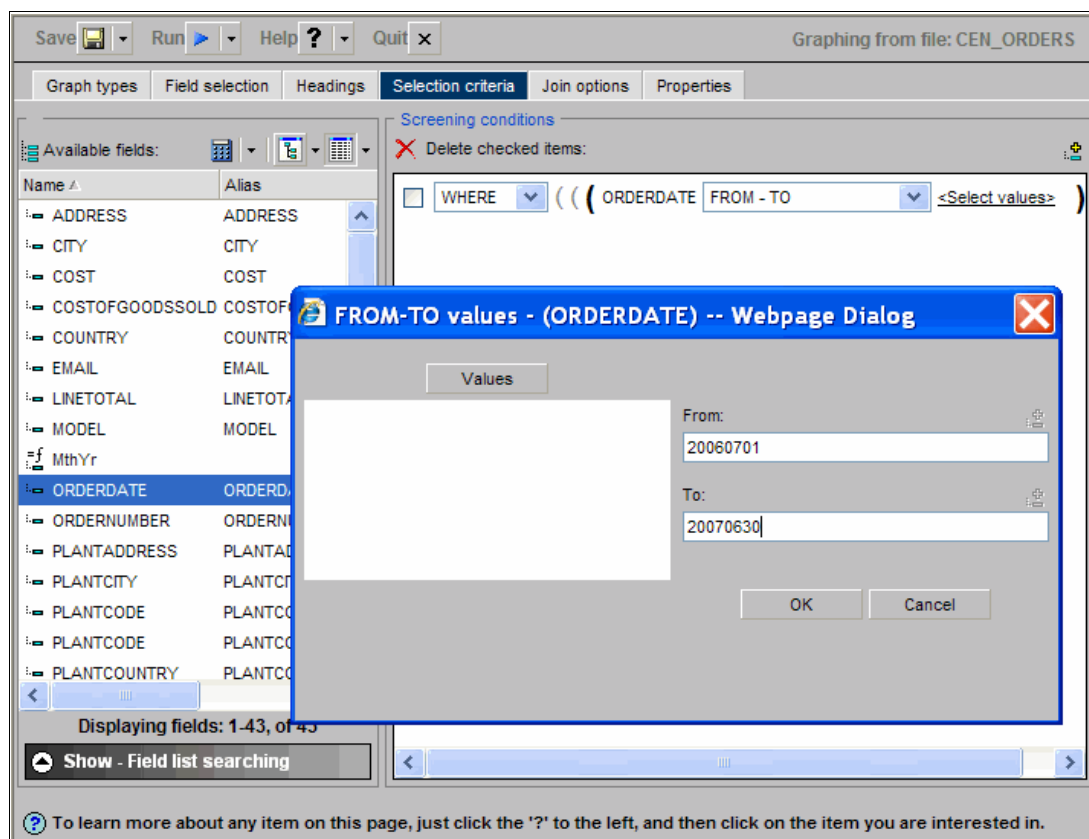


Figure 6-39 Date From-To option

- v. Run your report. Again the same report is generated.

6.5.2 Adding a user-specified date range parameter

In this next example, we allow the user to specify their date range instead of forcing one on them:

18. Remove your existing Screening conditions.

19. Drag **ORDERDATE** to the Screening conditions panel.

20. In the Screening conditions pane (Figure 6-40), complete these steps:

- a. Select **GREATER THAN or EQUAL to** and click **Select values**.
- b. In the GE Values window (Figure 6-40), instead of specifying a constant, select **Parameter**. Enter a parameter name for your user such as FROM_DATE. Click **OK**.

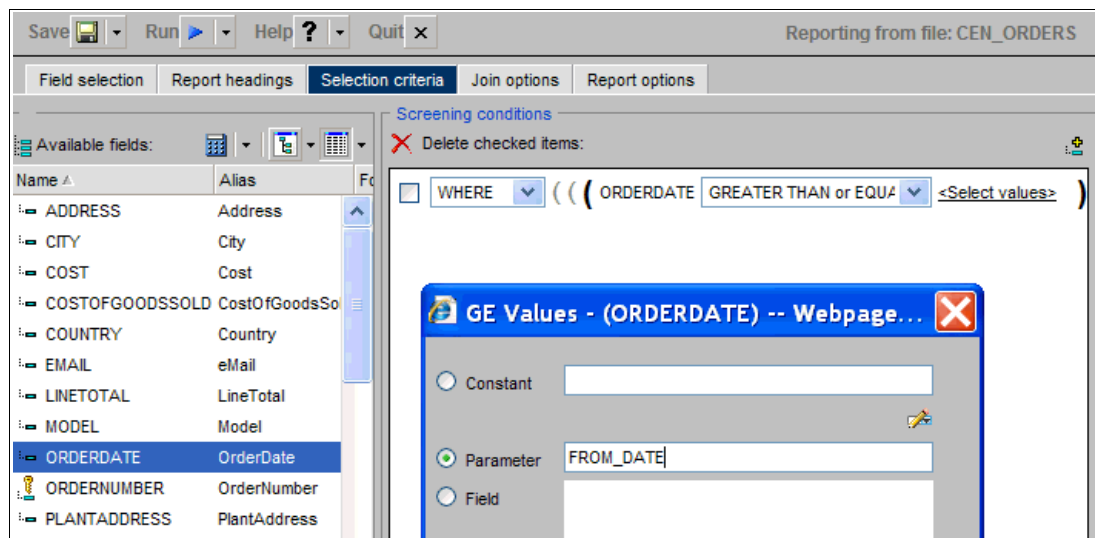


Figure 6-40 Creating a date parameter

21. Again drag **ORDERDATE** to the Screening conditions pane.

22. In the Screening conditions pane (Figure 6-41), complete these steps:

- a. Select **LESS THAN or EQUAL to**. Click **Select values**.
- b. In the LE Values window, select **Parameter** and name the parameter TO_DATE.

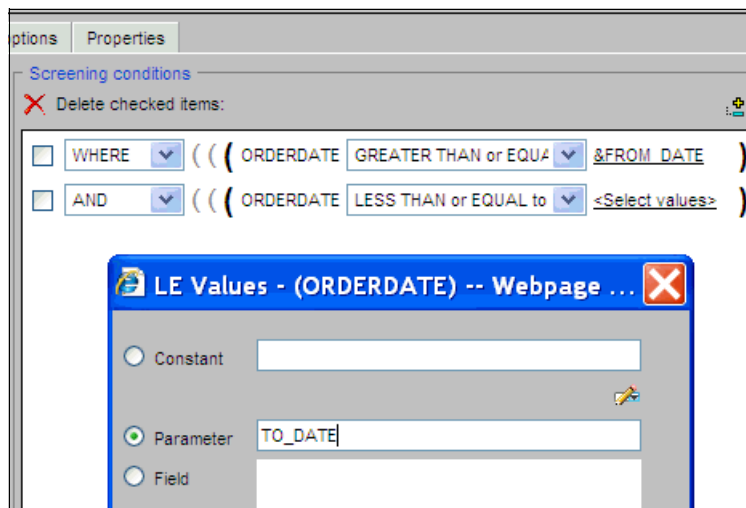


Figure 6-41 Defining parameters for a date range

23. Save and run your report. Figure 6-42 shows the results of running this report.

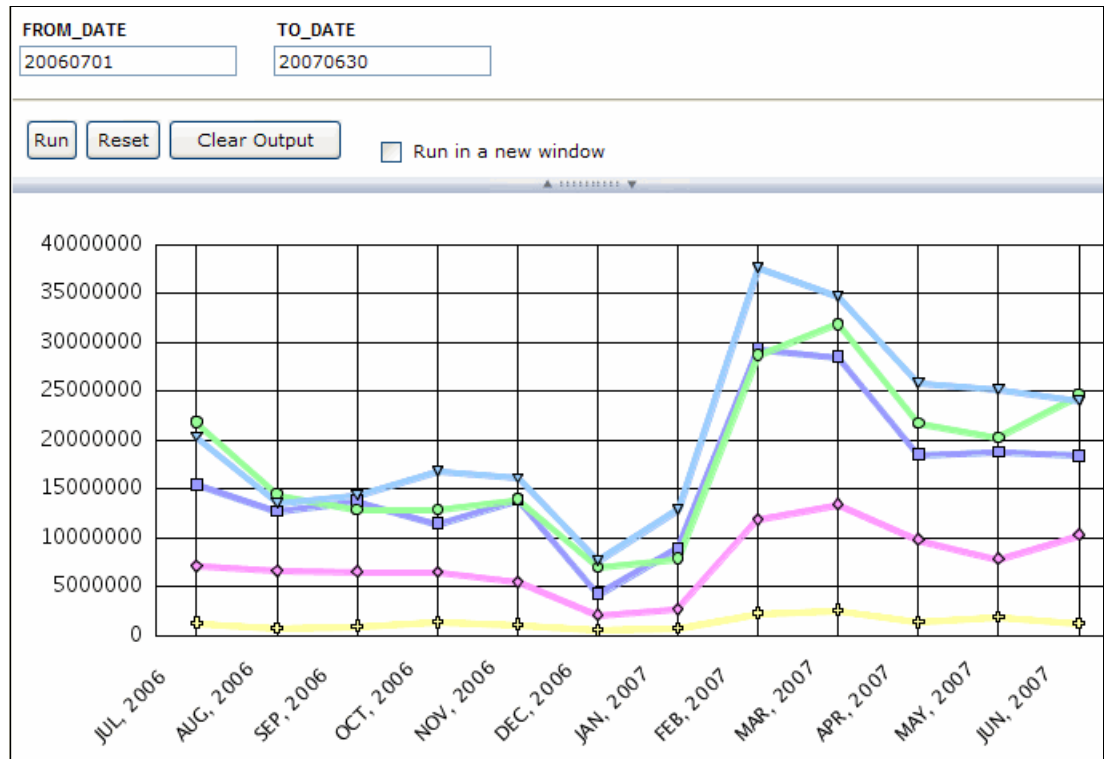


Figure 6-42 Trend graph with variable date range

24. Try various date combinations.

In this exercise, we saw multiple ways to hard code a date range and how to have the user specify a variable date range.

6.6 Working with pie charts (GA4_Pie)

In this exercise, we create a single chart that contains multiple pie charts, one for each year. We then add conditional styling or traffic lighting to our pie charts based on comparisons with calculated values. See Figure 6-43.

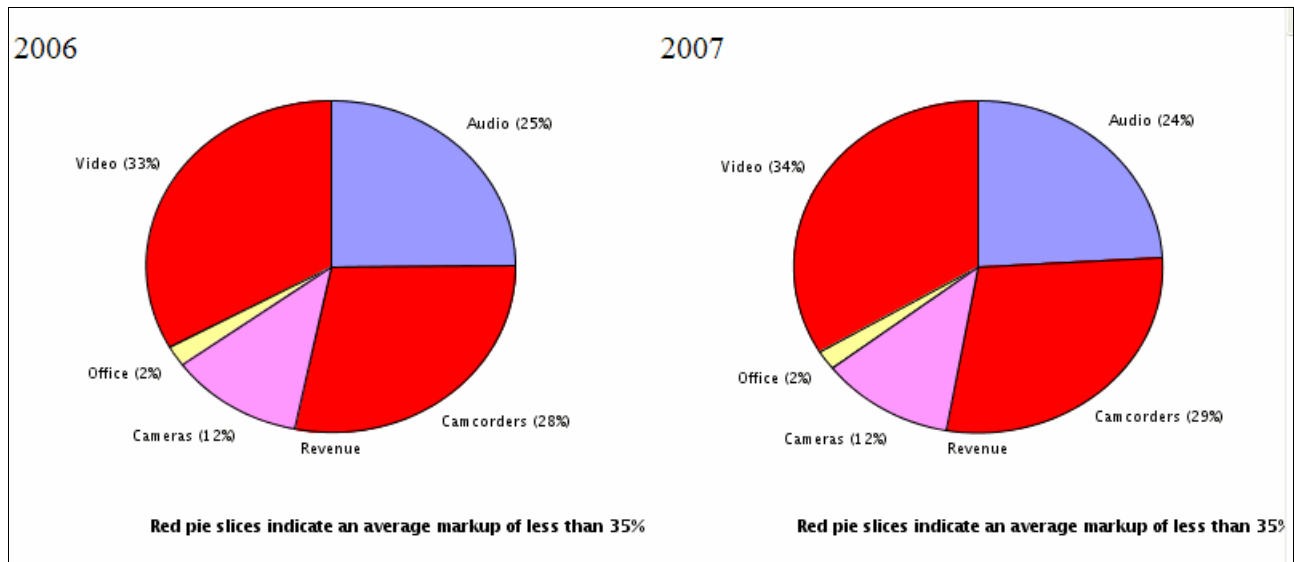


Figure 6-43 Yearly pie charts with conditional styling or traffic lighting

6.7 Creating a product type pie chart for each year

To create a product type pie chart for each year:

1. Navigate to your **Tutorials** folder, right-click and select **Graph Assistant**.
2. In the window that opens, choose the **Cen_Orders** table and click **OK**.

3. On the Graph types tab (Figure 6-44), under Select from the following graph types, select **Pie**. For Select from the following Graph Styles, select **Basic**.

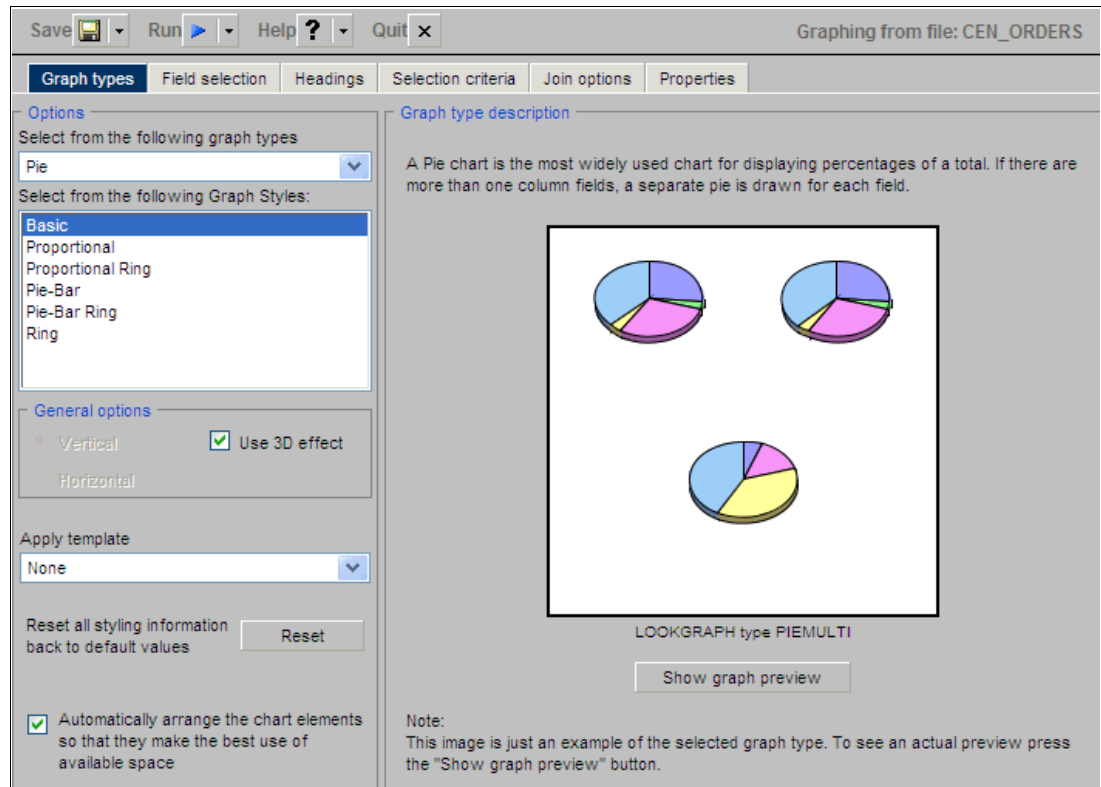


Figure 6-44 Selecting a graph type

In this example, we create a pie chart for each year in our data. We do not currently have a field that contains just the year.

4. Click the **Field selection** tab.
5. Click the **calculator** icon and select **New define field**.

6. In the Define field creator window (Figure 6-45), for Field, type Year and for Format, type YY. Remember that this creates a four-digit year, for example 2007. In the pane under Field, specify that the field is to be based on ORDERDATE. Click **OK**.

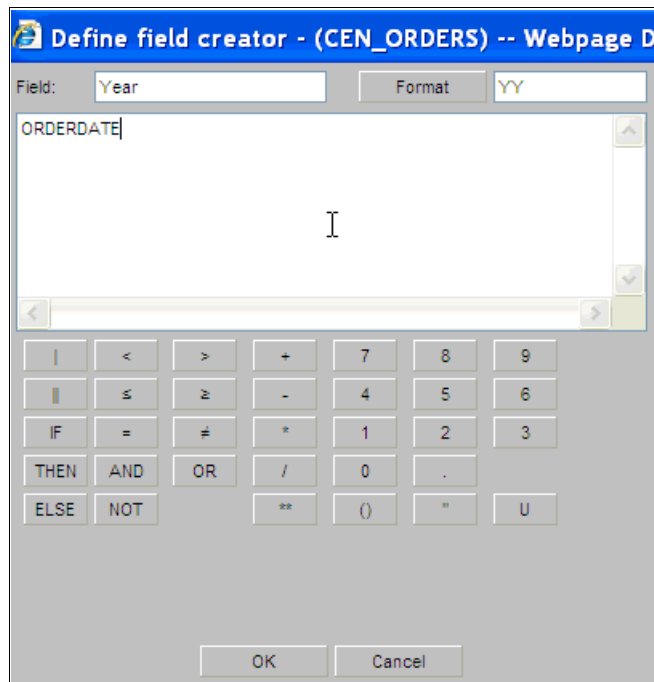


Figure 6-45 Creating a year field

Tip: If you are tired of splitting ORDERDATE into its various components for each report, make sure that you continue to progress to the tutorial in 12.1.1, “Date Decomposition” on page 269. This tutorial shows how to automatically break down a date and make the components available to all reports.

7. Click the **Field selection** tab to define the fields used in the pie chart. See Figure 6-46 on page 143.
8. Drag **PRODUCTTYPE** to the Field value plotted on X axis pane. This means that we will have a pie slice for each product type.
9. Drag **LINETOTAL** to the Field value(s) plotted on Y axis box. This means that the product type pie slices will be proportionately sized based on the LINETOTAL summed values.
10. Drag **Year** to the Create a separate graph for each value of this field pane. By default, this creates a separate graph for each value in Year.

11. Click **Show - Field options - Year**.

- a. Select the **Display graph in number of columns** option.
- b. Type a value of 2 next to the Display graph in number of columns option to indicate that you want two graphs across the page. If you had six different years, these settings would create six pie charts in a 2 x 3 layout. If you specified six across, you would have six, small pie charts across the page in a single row.

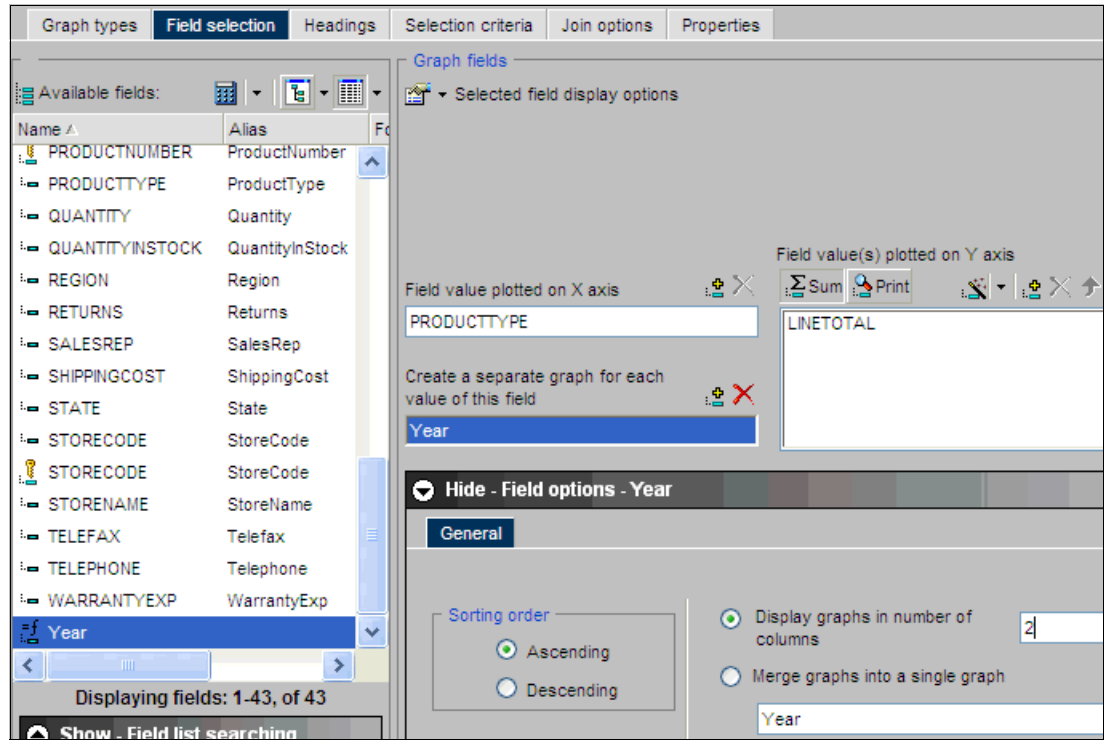


Figure 6-46 Pie chart field definitions

12. Run or preview your chart. The graph is essentially complete.

13. Often a graph is easier to read without a legend. Instead we want the labels directly on or next to the pie slices for easy readability. Click the **Properties** tab and the **Pie** subtab (Figure 6-47).

- a. In the Graph preview window, select the **Automatic update** option. Whether you permanently work in Automatic update mode is strictly a performance question.

Tip: If you closed or lost your Graph preview window, click the **Graph types** tab and select **Show graph preview**.

- b. While viewing the Graph preview window, select **Show pie labels**.
- c. Try selecting the different option combinations from both columns to see the results. In the end, select the **Labels only** and **Slice label and % value** options.
- d. In the Graph preview window, clear the **Automatic update** option.

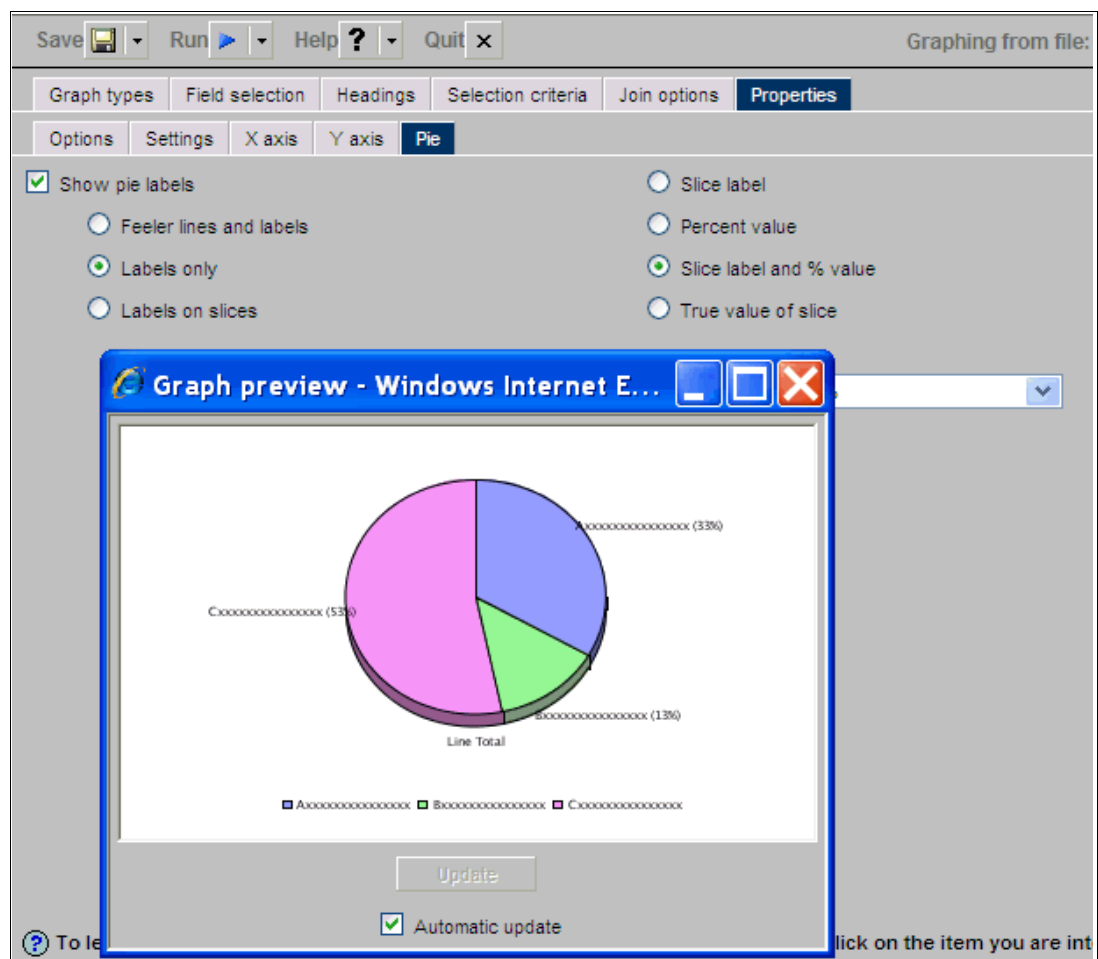


Figure 6-47 Automatic update for Graph preview

14. As you can see in the Graph preview, you still see the legend, which duplicates the information on the slice labels. To remove the legend from your pie chart, click the **Properties** tab and the **Settings** subtab.

15. On the Settings tab (Figure 6-48), clear the **Show legend** option.

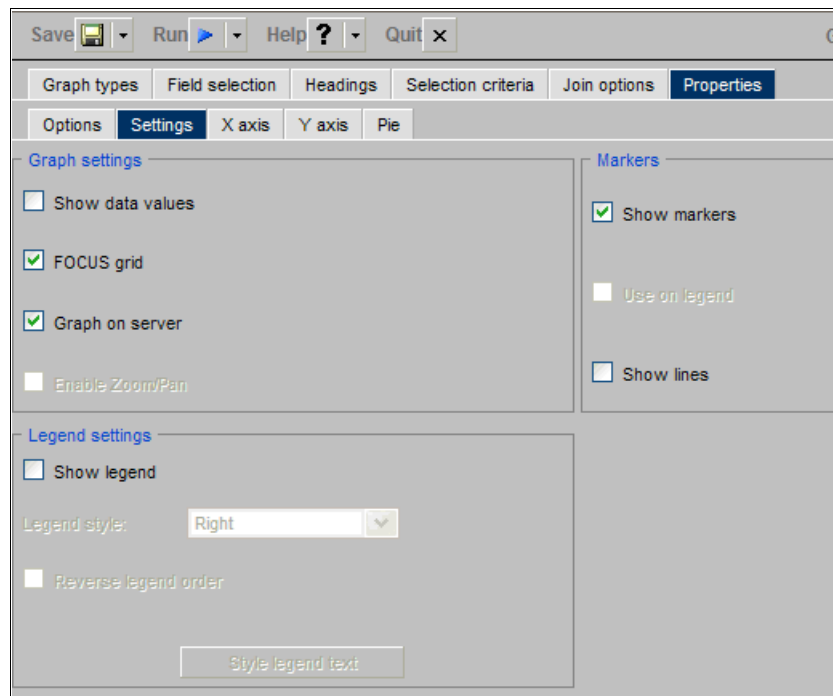


Figure 6-48 Legend removal

16. Tell DB2 Web Query the size to make each of our pie charts:

- a. Click the **Properties** tab and the **Options** subtab.
- b. On the Options tab, select **Custom size**. For Horizontal, type 400 pixels, and for Vertical, type 300 pixels.

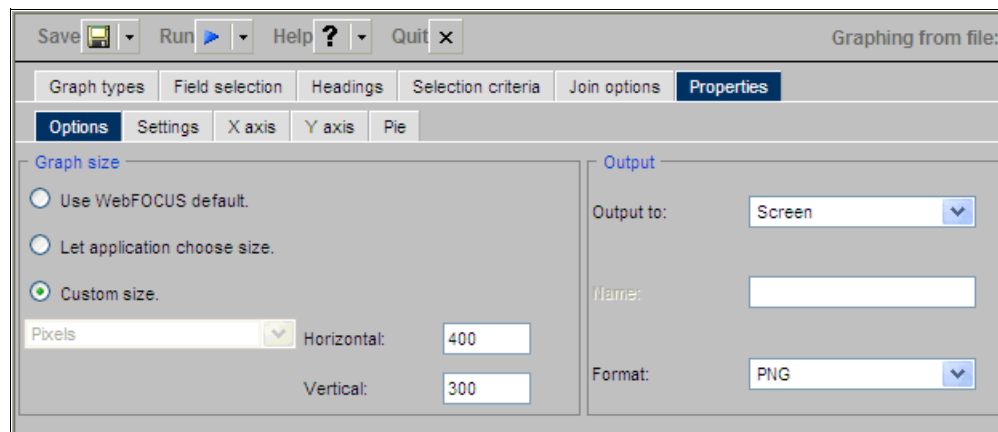


Figure 6-49 Specifying custom graph sizes

17. Save your chart as GA4_Pie and run the report. Figure 6-50 shows the results of running this report.

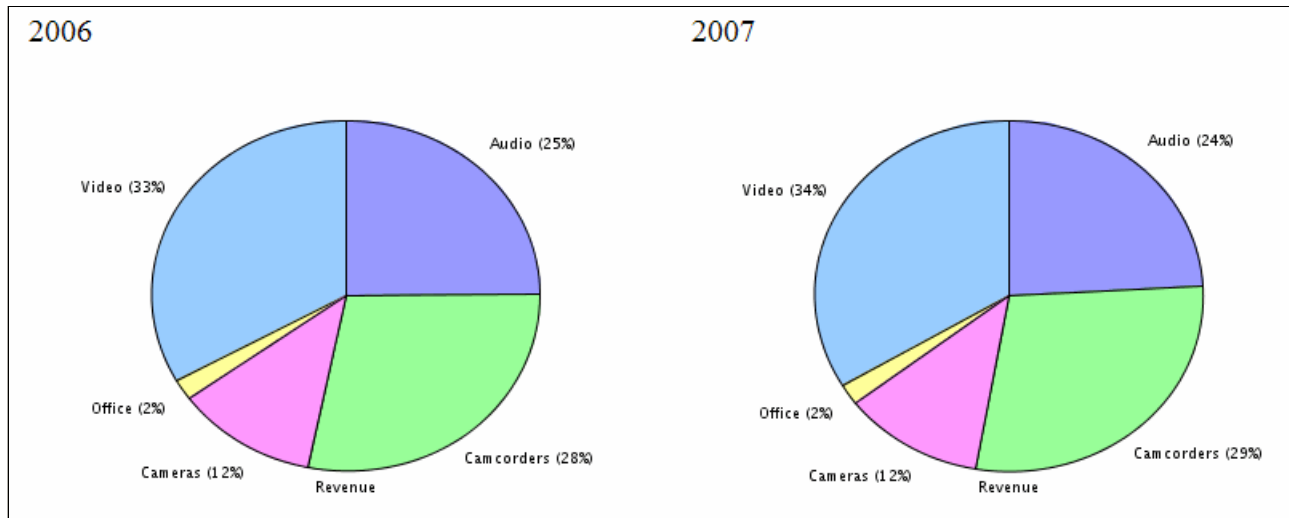


Figure 6-50 GA4_Pie chart

As we expect, notice how similar the graphs look compared to the Graph preview window in Figure 6-47 on page 144.

6.8 Conditional styling or traffic lighting a graph

The last area that we want to address in our graphing tutorial involves having segments of the graph highlighted based on a condition in our data. It is straight forward to say highlight LINETOTAL when LINETOTAL is greater or less than a predefined number. In this case, we want to highlight the pie segments based on the profit margin of that segment. Specifically if the markup that the product is sold for is lower on average than corporate standards, we want to highlight this.

We must calculate the average markup after we aggregate our data. Summing an average calculated at record input time gives us inaccurate results. As previously discussed with DB2 Web Query, you can specify whether you want a new field calculated as records are read or after the input records are processed and aggregated. This processing sequence is described in more detail in Appendix C, "Processing differences between the define and compute fields" on page 375.

Note: DB2 Web Query calls a new field calculated at input time a *define field*. A field that is calculated after all aggregation is done is called a *compute field*. See Figure C-1 on page 375 for more details.

1. If you are still in your GA4_Pie report, click the **Field Selection** tab (Figure 6-51).
If this report is not open, open the GA4_Pie report. On the DB2 Web Query home page, right-click the **GA4_Pie** report and select **Open**.
2. To create a compute field, select the **Wizard** icon next to the Print button.

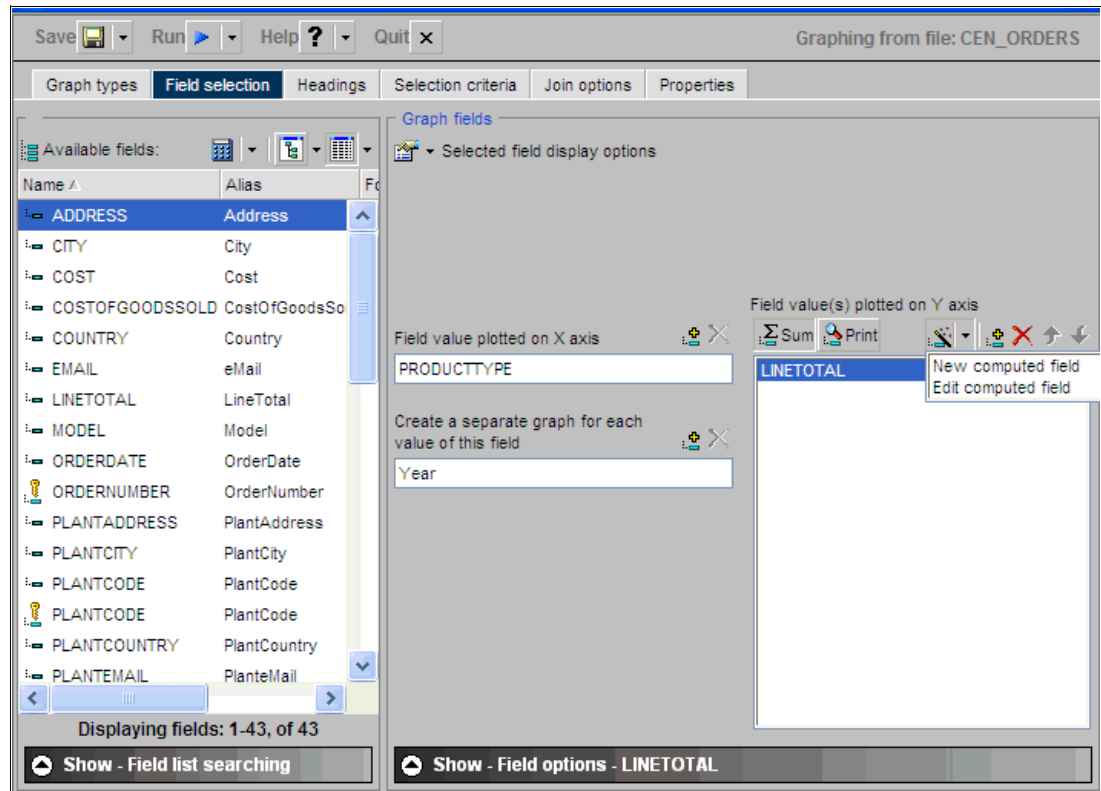


Figure 6-51 Initiating the compute field wizard

3. In the Compute field creator window (Figure 6-52), for Field, type MARKUP. Specify the formula in the pane below Field. Type the following text in the pane:

$((\text{LINETOTAL} - \text{COSTOFGOODSSOLD}) / \text{COSTOFGOODSSOLD}) * 100$

You can either type the field names or select them by double-clicking the names on the left. The DB2 field names must be in uppercase.

Click **OK**.

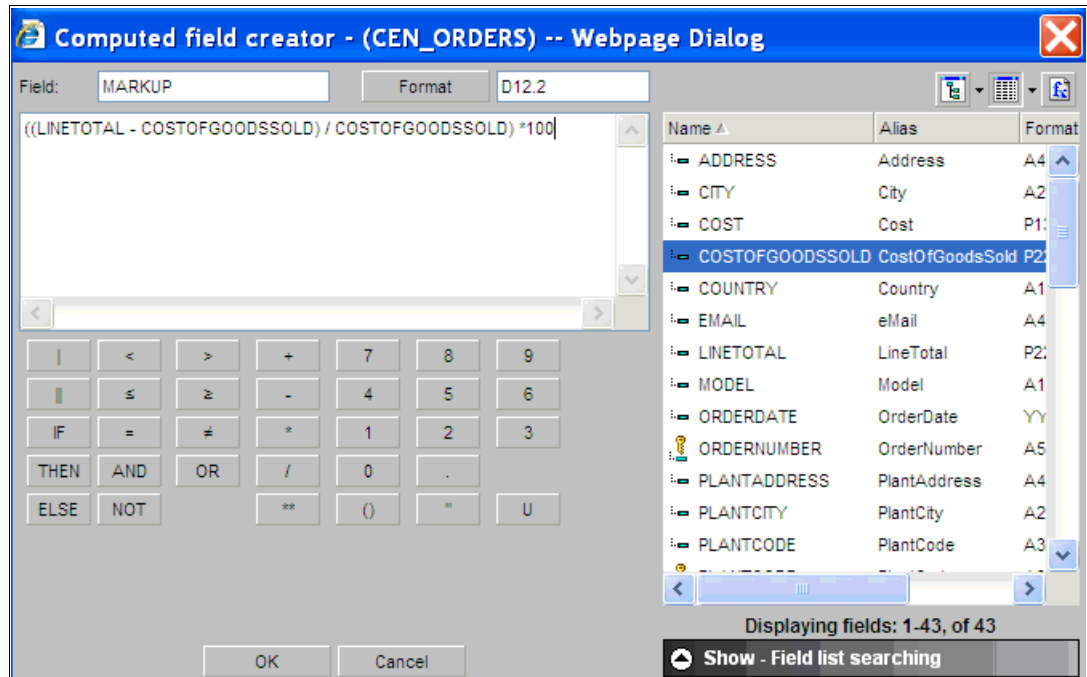


Figure 6-52 Creating a compute field

While MARKUP is a field that we need to reference in our conditional styling, we do not want to display a graph of MARKUP data.

4. Under Field value(s) plotted on Y axis, highlight **MARKUP**.
5. Click **Show - Field options** (Figure 6-53). On the Display tab, select the **Make this field invisible** option. This makes MARKUP available to the report but does not display it on the graph.

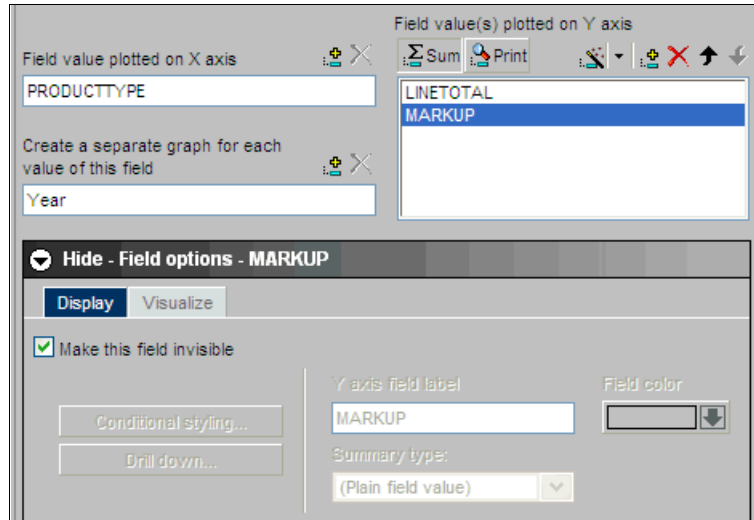


Figure 6-53 Making a field invisible

6. Define the pie slices that you want to highlight (Figure 6-54):
 - a. Highlight **LINETOTAL** under Field value(s) plotted on Y axis.
 - b. Click **Show - Field options**.
 - c. Click the **Conditional styling** button.

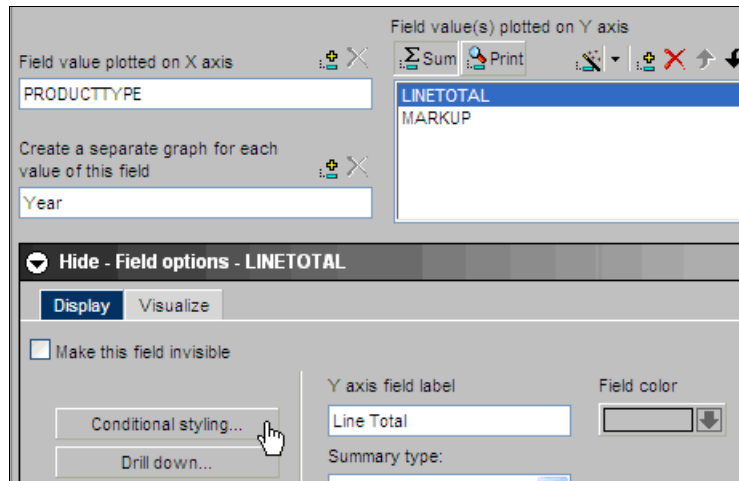


Figure 6-54 Conditional styling

- d. Click the **Add** wizard icon (Figure 6-55).

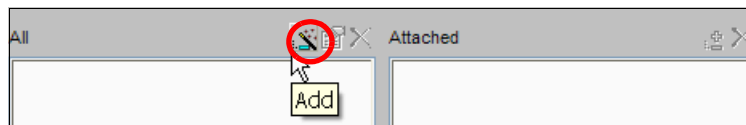


Figure 6-55 Conditional Styling Add wizard

- e. We want to highlight the pie segments or slices that have a lower markup than our company standard of 35% (Figure 6-56).
 - i. Give this condition the name of Low_Markup.
 - ii. For Field, select **MARKUP**.
 - iii. For Relations, select **is less than or equal to**.
 - iv. For Please select a value, type the value of 35.
 - v. Click **OK**.

Figure 6-56 Creating a conditional styling condition

- f. Attach the condition definition to LINETOTAL and define what happens when this condition is satisfied.
 - i. Highlight **Low_Markup** and click the **Add Condition** icon. See Figure 6-57.

Figure 6-57 Attaching a condition definition

- ii. Click the grey color indicator box (circled in Figure 6-58 on page 151).

- iii. In the Color window (Figure 6-58), select the color that you want the pie slices when the slice meets the Low_Markup condition. In this example, choose **Red**.

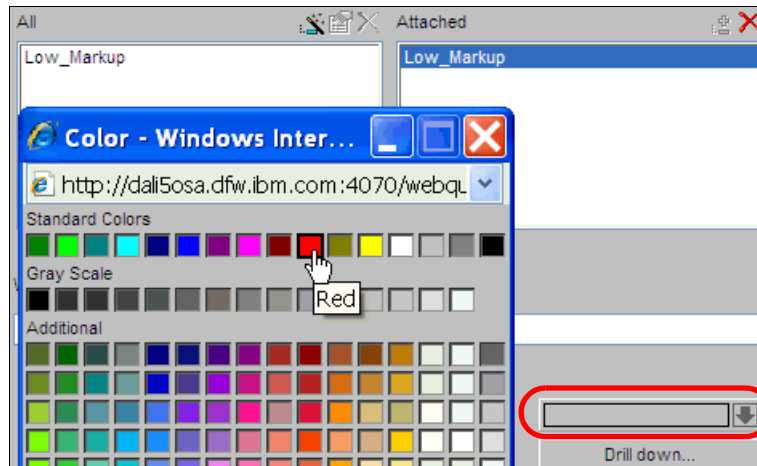


Figure 6-58 Conditional styling highlighting

7. When modifying a report by applying filters or using traffic lighting, we recommend that you indicate this information on the report itself. You typically do this via a subheading or a page footing or footnote.
 - a. Select the **Headings** tab and click the **Graph** subtab (Figure 6-59).
 - b. Under Chart footnote, type Red pie slices indicate an average markup of less than 35%.

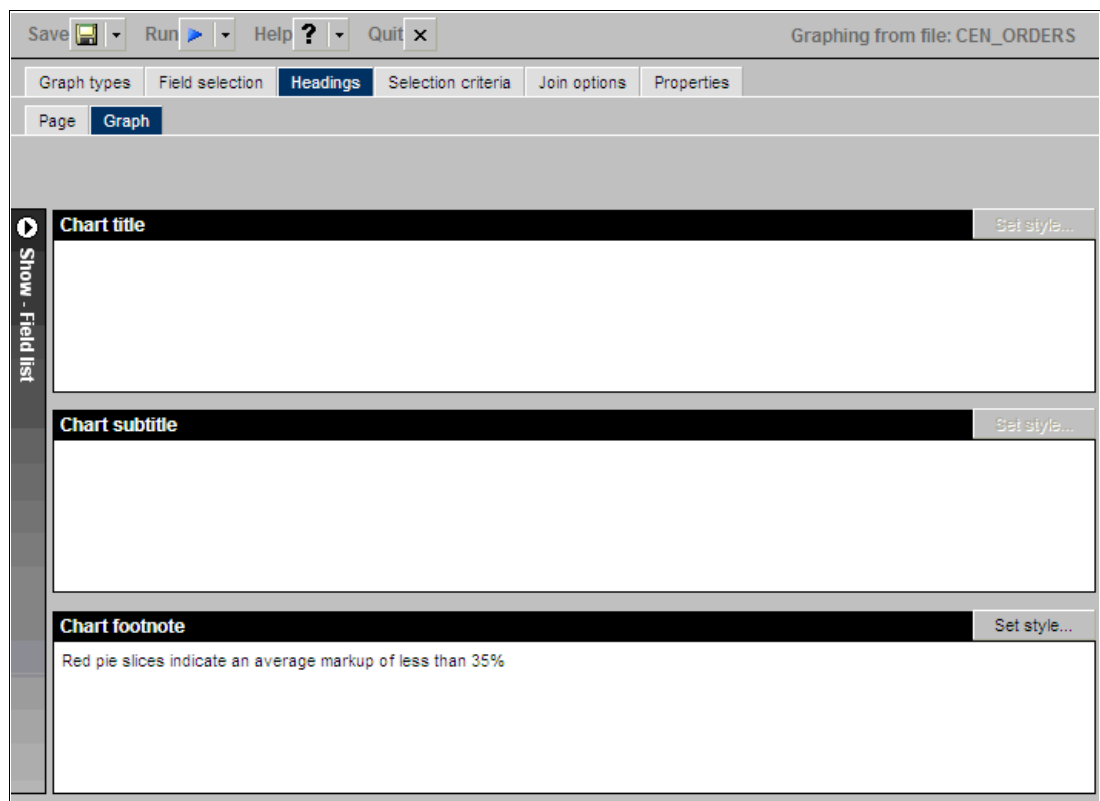


Figure 6-59 Explanatory chart footnote

8. Save your report and run it. Figure 6-60 shows the results of running this report.

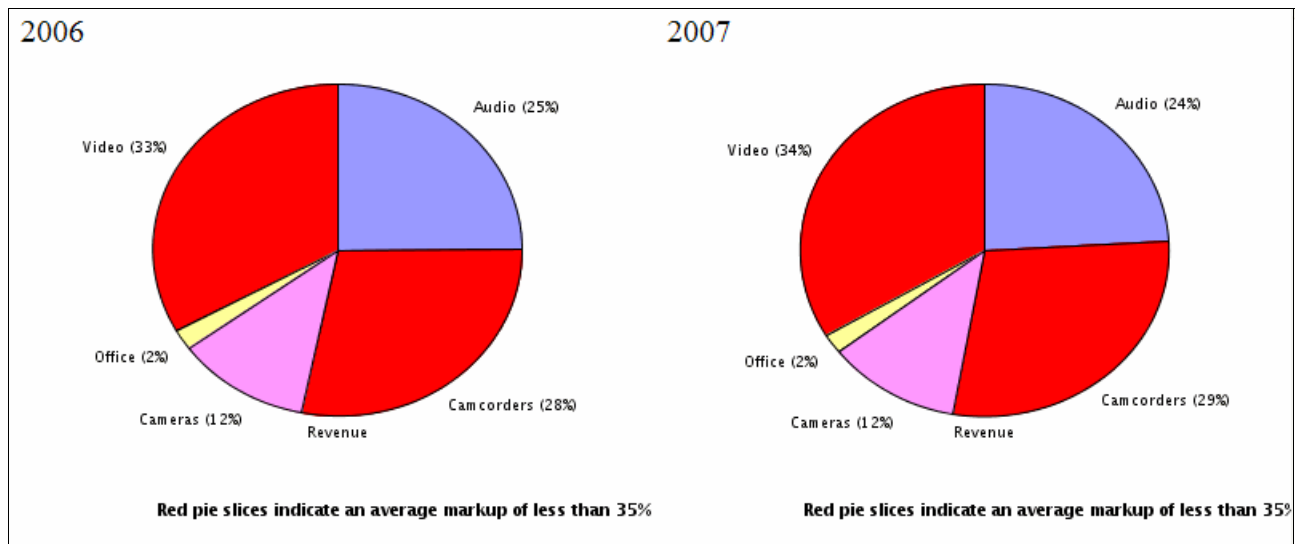


Figure 6-60 GA4_Pie low markup charts

You have now finished the Graph Assistant tutorial.



Power Painter

Power Painter is a Web layout and report creation tool. It is built using the new AJAX technology. This allows a thin browser to deliver advanced functionality without add-ons and plug-ins. With HTML, you must update the entire panel. With AJAX technology, the application can update portions of the panel for improved flexibility and performance. Power Painter allows you to create a compound report combining multiple graphs and charts on a single page. You can link the various charts together all tied to a common sort group. Power Painter supports a live feed to your data to allow for real What You See Is What You Get (WYSIWYG) report creation. With Power Painter, you can use multiple data sources to build your compound reports.

In this tutorial, we build reports similar to those that we built using Report Assistant and Graph Assistant. The idea is for you to become familiar with the various ways of achieving similar end results. We lay out the report and graph on one page, link them by COUNTRY, and tailor the request for PDF output.

7.1 Tutorial overview (PP1_PDF)

In this tutorial, we build a PDF file that shows profit analysis by region with a separate graph and report for each country. The PDF file has a table of contents that lets a user quickly jump to the country that they are interested in. See the example in Figure 7-1.

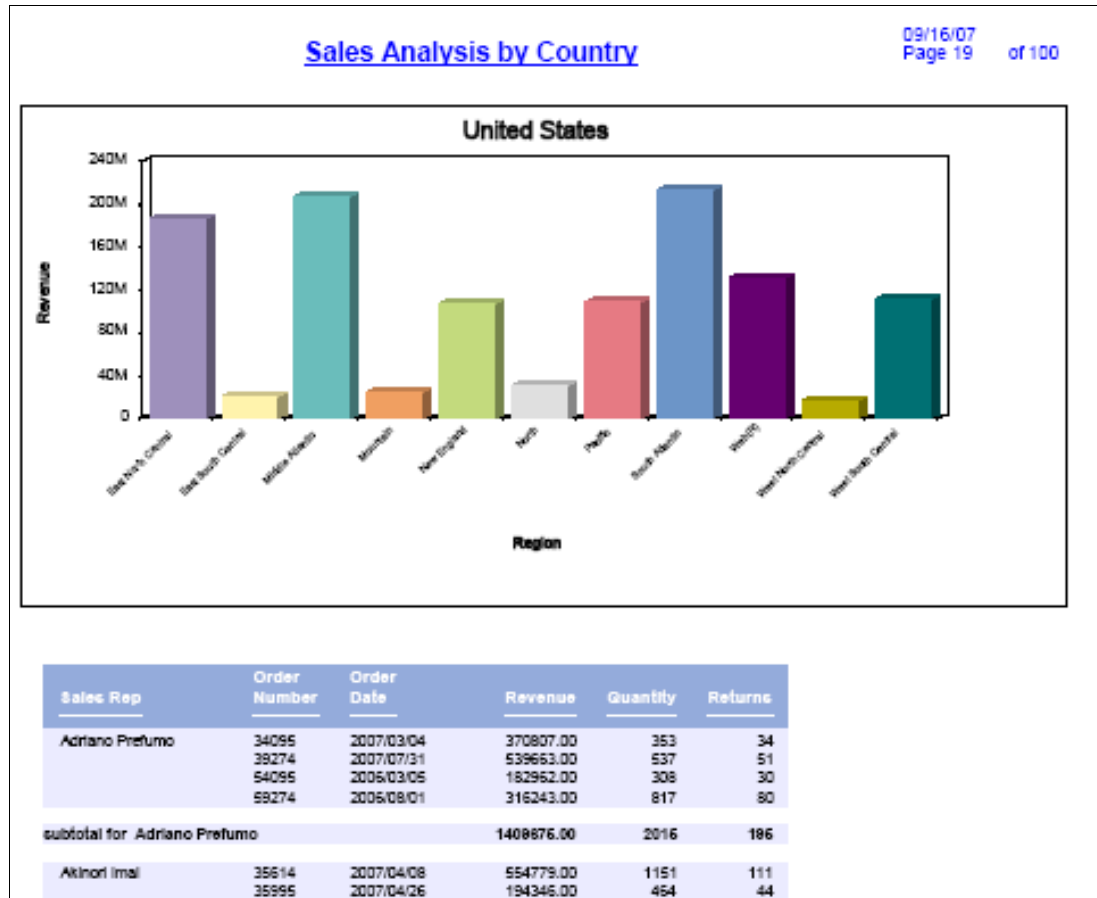


Figure 7-1 Power Painter exercise PDF output

7.2 Launching Power Painter and tailoring the main window

In this section, we explain how to launch Power Painter and tailor the main Power Painter window to look and work in a way that we are comfortable with. We do this before we create our report.

The main Power Painter window is composed of standard drop-down menus at the top (File, Edit, and so on), a row of icons immediately below the menus, an empty canvas on which to design your report, and various palettes. You can move the palettes around the window or you can dock them on the right side. To open the palettes, you can use **View** → **Palette** → **various palettes**. You can open most of the palettes by using the icons below the menus. To know what the function is of each icon, hover over an icon to see a description.

Let us get started:

1. Open DB2 Web Query, if it is closed.
2. Navigate to your **Tutorials** folder, right-click, and select **Power Painter**.
3. In the Open window (Figure 7-2), select the **Cen_Orders** table and click **OK**.

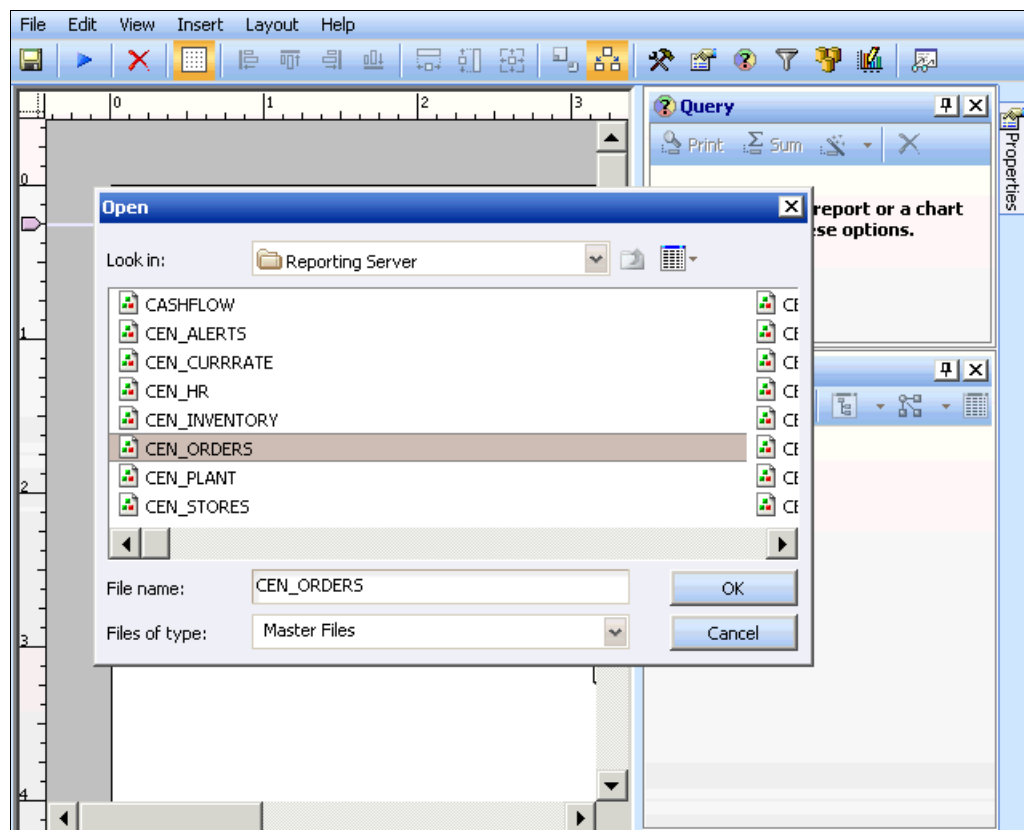


Figure 7-2 Selecting a data source in Power Painter

4. Customize the window:
 - a. Double-click a palette or move it to the edge of the canvas to dock it. You should see the pushpin and close (X) buttons in the upper right corner when a palette is docked. To undock the palette, drag it to a new location on the window.
 - b. Click the **pushpin** button to minimize the palette and have it appear as a tab on the right side of your window. Hovering over the tab makes the palette reappear.

You will want to change which palettes are open based on the work that you are doing. The example in Figure 7-3 is one design that appears to work quite well and will be used in this tutorial.

- Click the **pushpin** button of the open Properties palette. This temporarily makes it a tab on the right side. We bring it back to the main docking area later.
- Either choose **View** → **Palettes** → **Query** or click the **query** icon (multi-colored circle with a question mark, circled in Figure 7-3).
- Drag the query palette to the right, underneath the Toolbox, and above your Data Sources palette.

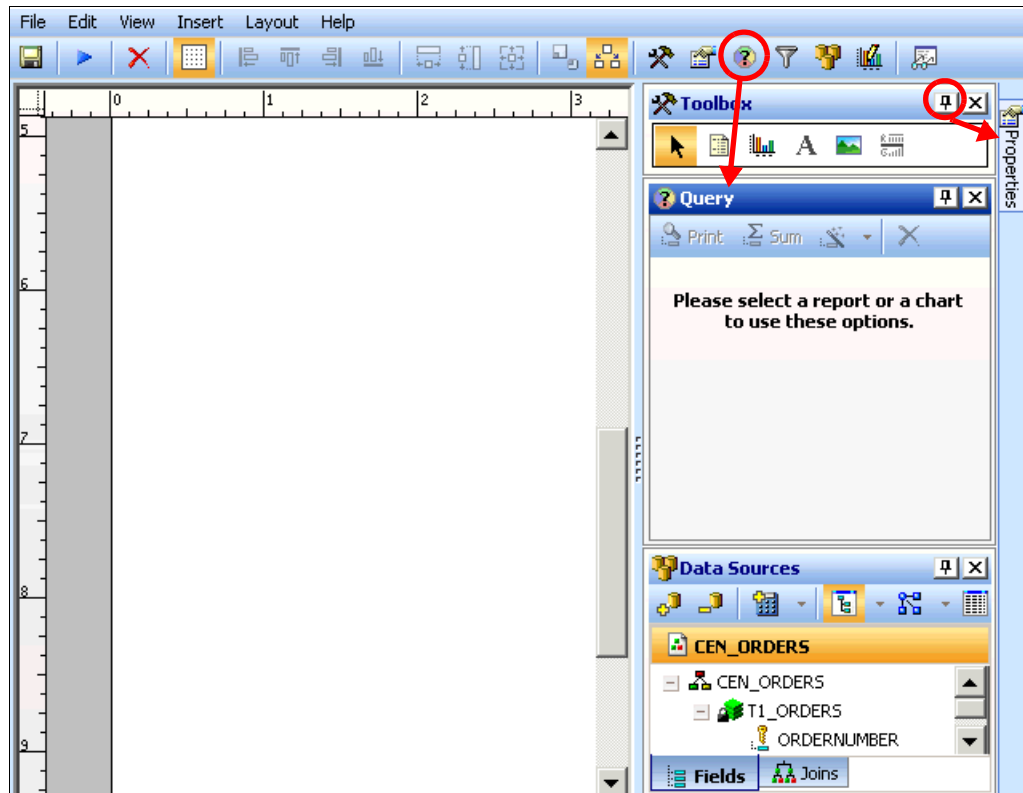


Figure 7-3 The main window in Power Painter

While you initially lay out your report, the two main palettes that you use are the Query palette and the Data Sources palette. After you select your columns, you can move the Properties box back to the docking area and click the pushpin button for Data Sources so that it minimizes to the side. This keeps the Query and Properties palettes in full view while you design the desired layout and design of your report. You can have more than two palettes open and docked, but they can become a little small to work with effectively.

In this report, we want our page to start with our logo or picture and a heading, with a graph below the logo and heading, and the main report below the graph.

7.3 Building a graph with Power Painter

The graphing component in Power Painter is a more full function and advanced graphing tool than Graph Assistant. It allows you more control over styling, colors, text properties, legend formatting, and more. Using gauges becomes much more meaningful with Power Painter. You can selectively control the color of the quality bands and the gauge needle. With Power Painter, you can add annotations to your graphs. While Graph Assistant allows you to drill down to another report or graph, this function is not available in Power Painter.

We start by building a bar graph that shows the individual regions in a country and their revenue attainment:

1. If you closed Power Painter, open it as explained in 7.2, “Launching Power Painter and tailoring the main window” on page 154.
2. Maximize your window.
3. Click the **graph** (bar chart) icon in the Toolbox (Figure 7-4).
4. Using your mouse, draw a rectangle on the canvas. This rectangle outlines the margins for your graph. Leave room for a report title at the top of the page. You can move and size the graph by using the mouse after you initially place it.

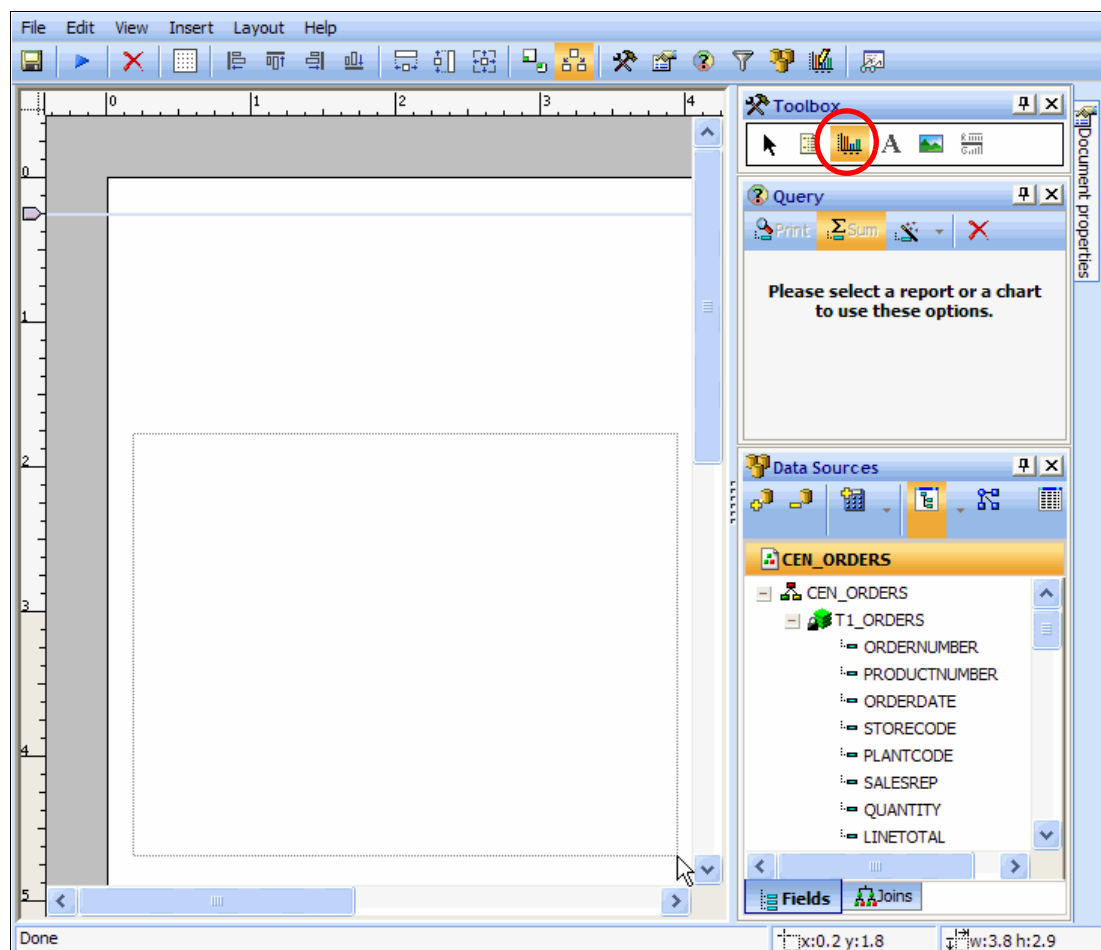


Figure 7-4 Inserting a graph

5. In the New Chart window (Figure 7-5) that opens, you see that Power Painter offers more advanced graph controls in most areas than Graph Assistant. For this example, select the default **Vertical Clustered Bar** chart.

If we want to display a series of bar charts, for example, one bar for each year for each representative, then we select the CrossTab box. However, we do not need to do this for this example.

Click **Finish**.

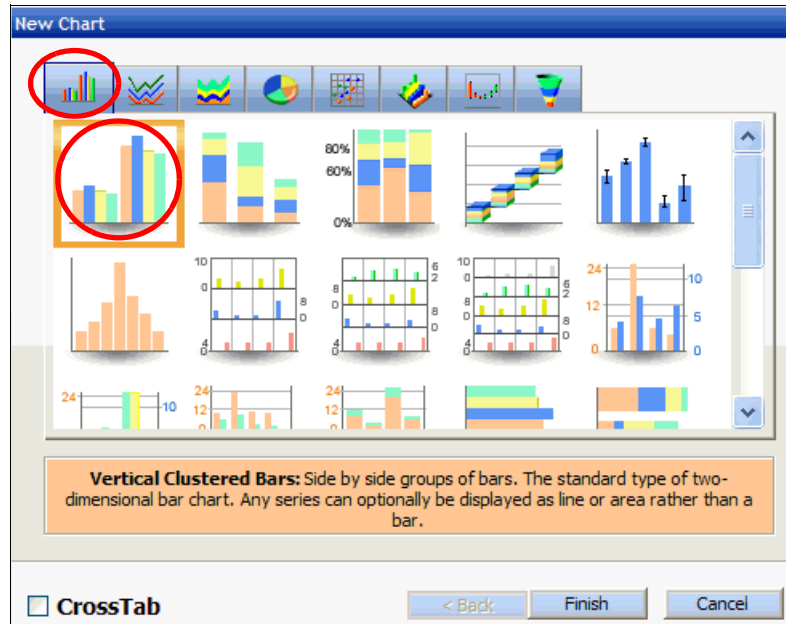


Figure 7-5 Power Painter selecting a graph type

6. Define the columns that are required in the graph.

Similar to Report Assistant and Graph Assistant, you can view column names in multiple sequences, including in alphabetical sequence, or in a tree type hierarchy based on table names. Remember that our Orders data source is composed of multiple tables that are all joined automatically by foreign key definitions. The icons in the Data Source bar allow you to change the displayed sequence.

Use the drop-down arrows on both the hierarchy structure icon and the list icon to see the different choices.

7. To include a field on the report, drag the field name either onto the graph that you just created or onto the Query palette. When you drag a field onto the report area of the canvas, a Query palette opens.

- a. Drag **LINETOTAL** onto the Y Axis label. See Figure 7-6.

Tip: Make sure that you select your graph before you drag the field. The open Query palette relates to the highlighted object.

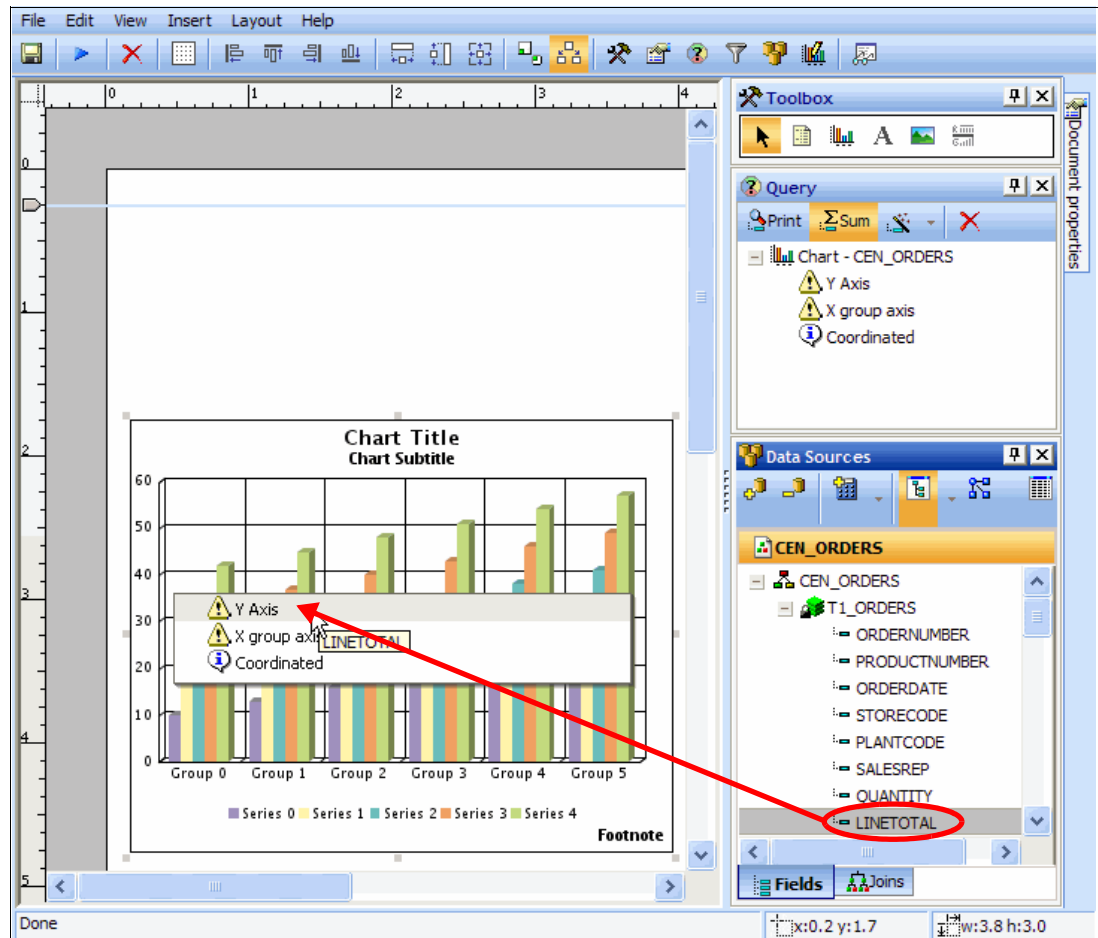


Figure 7-6 Field Selection for graphing

- b. Drag **REGION** onto the X group axis label (Figure 7-7).

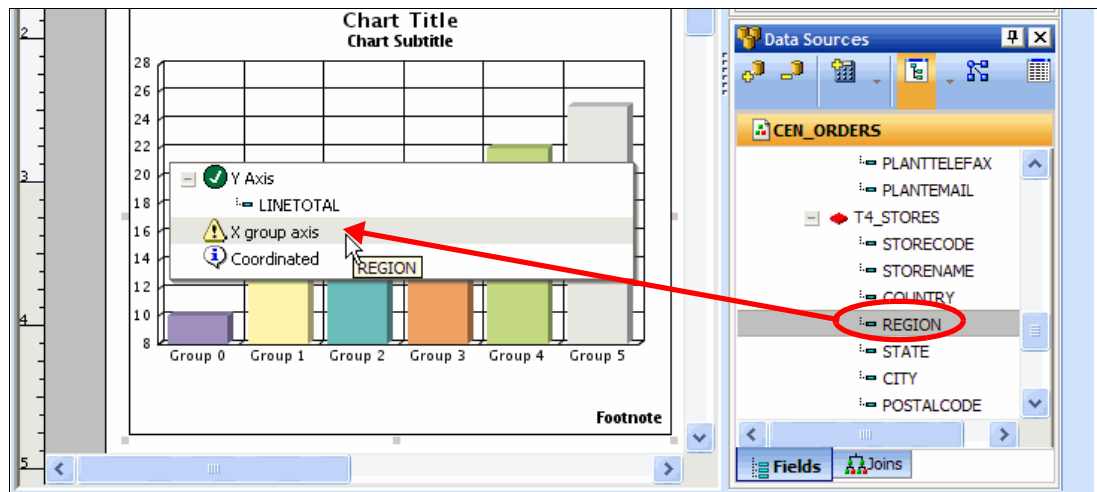


Figure 7-7 X group axis field selection

- c. Drag **COUNTRY** onto the Coordinated label (Figure 7-8).

Tip: Coordinated fields are common sort fields and filters that tie together the various reports and graphs on your page. Each change in the coordinated field causes a page break. The report layout you define is repeated for each new value.

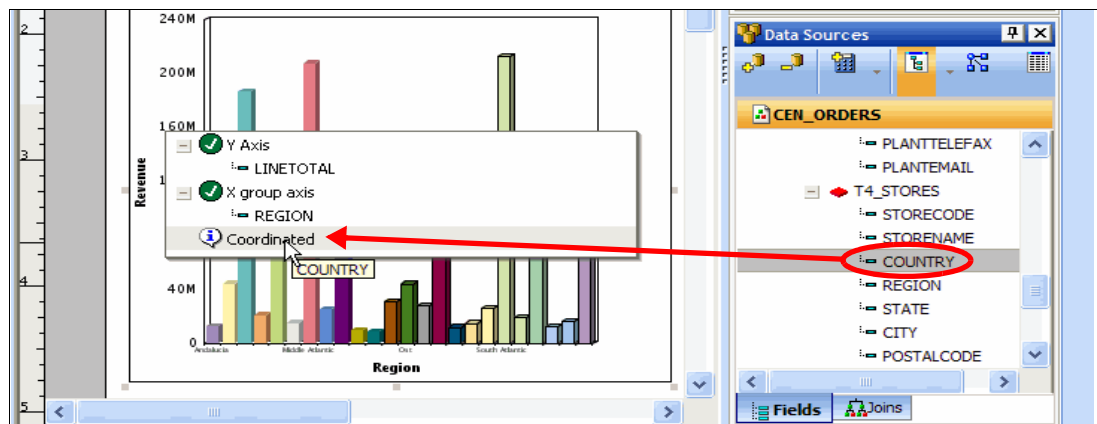


Figure 7-8 Adding a Coordinated field

8. Save your report as PP1_PDF.
9. After choosing most of the fields, you might find it easier to bring the Properties palette to the front and minimize the Data Sources palette:
 - a. Click the **pushpin** icon to minimize the Data Sources palette.
 - b. Double-click the **Properties** tab on the right side of the window to redock the Properties palette.

Now any time you select a field in the query palette, you see that field's properties in the Property palette.

Tip: The Property box changes between Document properties, Report properties, and Column properties. If you select a column in the Query palette, you see the properties that relate to that column. If you select a graph or report, you see the basic graph properties or report properties. And if you click the white space on the canvas outside of a report, you see the overall properties for the document.

10. Select a graph to open the Basic Chart Properties window (Figure 7-9).

To see all the chart properties, you must open the Chart Editor. Click either the **Show Chart Editor** icon or select **View** → **Palettes** → **Chart Editor**.

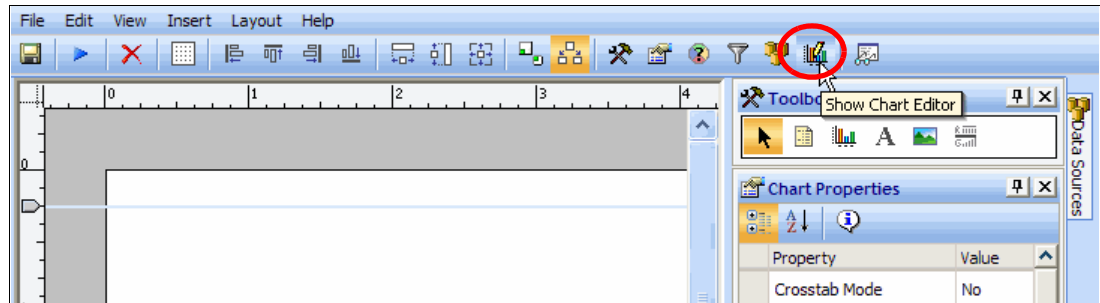


Figure 7-9 Show Chart Editor icon

11. With the Chart Editor (Figure 7-10 on page 162) open, the options that you select control the properties displayed in the properties panel. To read all of our region names, we must orient them differently on the graph:

- In the Chart Editor panel, select **Ordinal axis** and then select **Labels**.
- From the Properties panel, for **Text rotation**, under the Value column, select **45 degrees** to present the region names on a 45 degree angle.
- Look at the various properties available between the Chart Editor and the properties box. This is where you will completely control the appearance of your graph. You define your traffic lighting or conditional styling here. You can choose whether one series shows as bars while a second series is displayed as a line chart and much more.

Our graph is now complete.

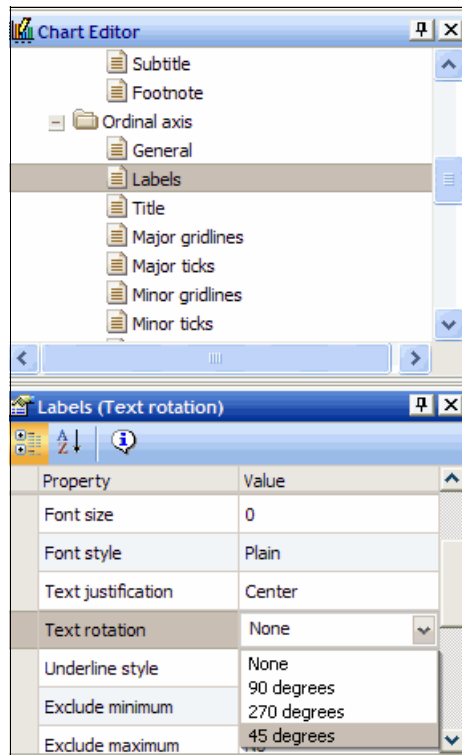


Figure 7-10 Chart editor properties

12. Add a heading to the report:

- a. From the Toolbox, select the **Text** icon (Figure 7-11).

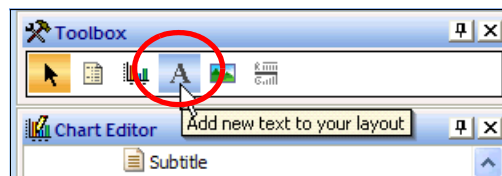


Figure 7-11 Adding text to a report

- b. Draw a rectangle where you want to place the heading of your report.
- c. Type a report title of Sales Analysis by Country.
- d. Click **Font Properties**.

- e. In the Edit Font window (Figure 7-12), complete these steps:
 - i. Choose a font; we choose **Arial**.
 - ii. Under Options, select **blue** for Text Color.
 - iii. Under Style, select **bold and underline**.
 - iv. Under Size, select **14**.
 - v. Click **OK**.

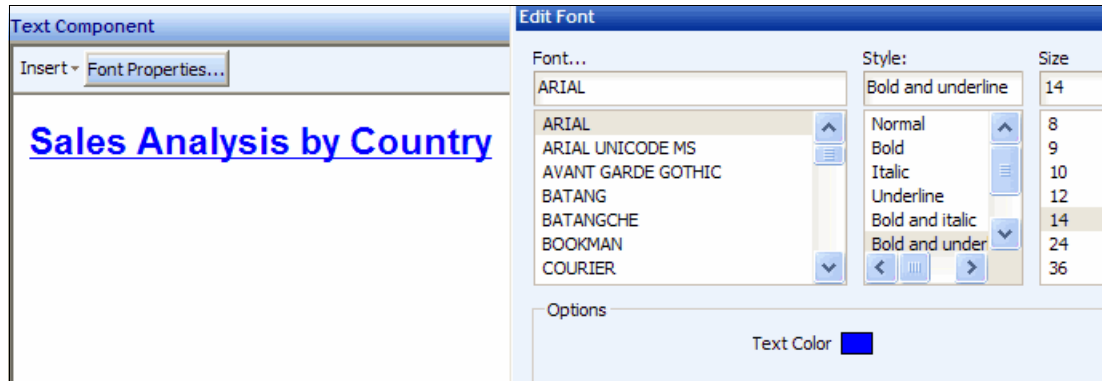


Figure 7-12 Adding text to a graph

13. Add a date and page numbers to the report:

- a. Select the **Text** icon from the toolbar and draw a rectangle where you want to place the text on your report (Figure 7-13).

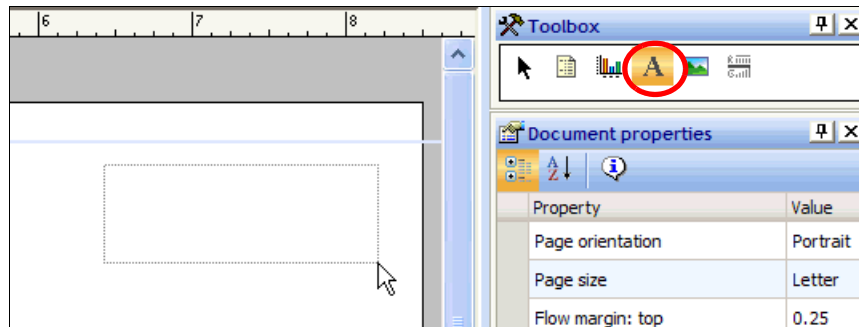


Figure 7-13 Adding a text box to your document

- b. Select the **Insert** → **Date/Time** and select the date format that you want to see (Figure 7-14).

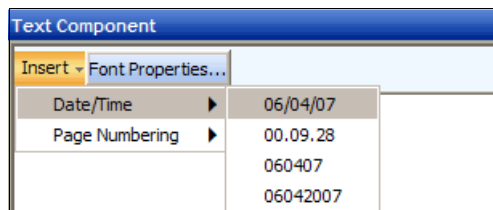


Figure 7-14 Inserting current Date/Time

- c. After the date that you just inserted, press the Enter key to force a new line and type Page.

- d. Select **Insert** → **Page Numbering** → **Page n of n** (Figure 7-15).

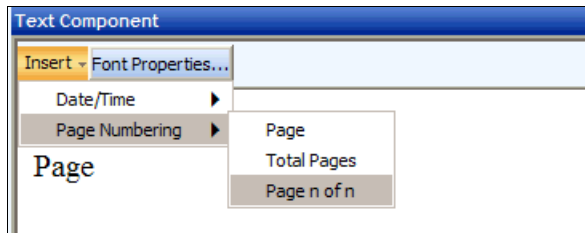


Figure 7-15 Inserting page numbering

14. Select the **Font Properties** button and make your new text the same color as your report heading (Figure 7-16).

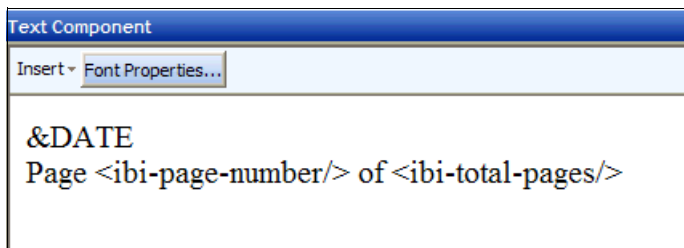


Figure 7-16 Date and page numbering area.

15. In the Edit Font window (Figure 7-17), change the font size to 10 so this text is smaller than the main heading. Click **OK**.

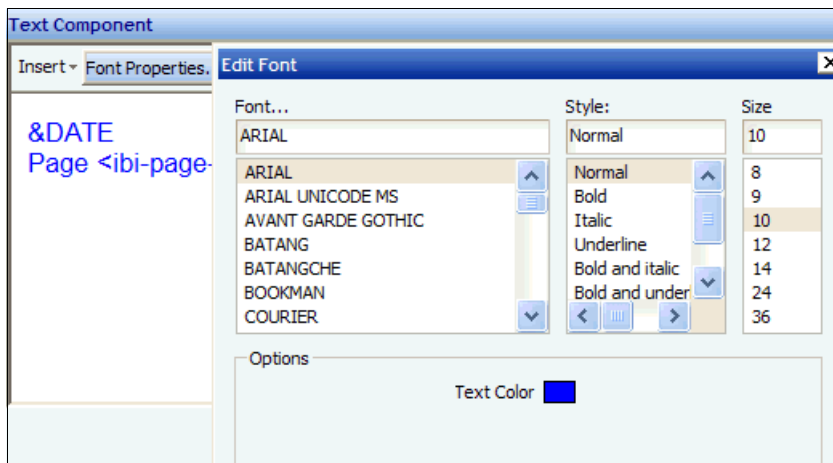


Figure 7-17 Font changes

16. Save and run your report. Figure 7-18 shows the results of running the report.

Notice that you have one page for each graph. When the coordinated column COUNTRY changes, a page break occurs.

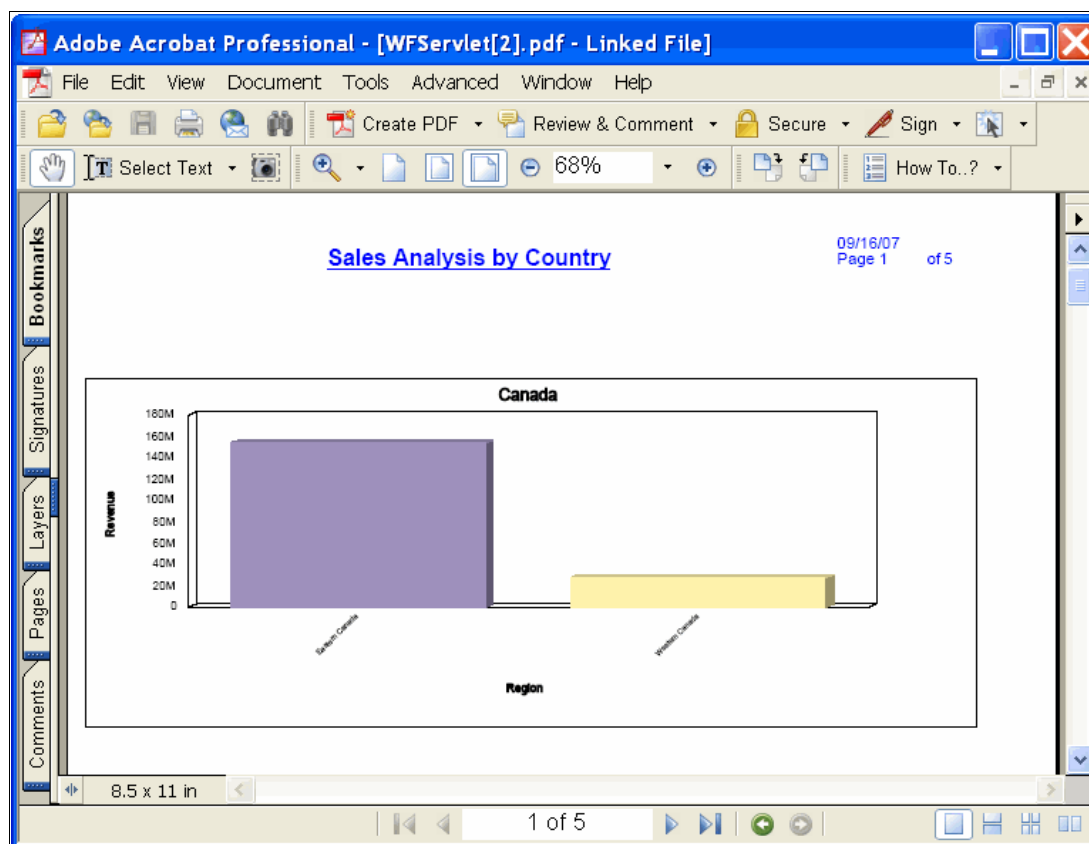


Figure 7-18 Regional Sales Analysis graph

Now that our graph is complete, we create the coordinated report as explained in the next section.

7.4 Building a report with Power Painter

We now add a report that lists each salesperson that sold in the country and details about the sales. It is conceivable that a salesperson's results are in different sections of the report if they made sales in different countries.

1. Make sure that your PP1_PDF report is open and maximized.

Note: We use the same data source that we used for the graph. If we want to add data from a second data source, we select the Data Sources palette and click the Add data source icon.

2. There are multiple ways to start a report in Power Painter. In this example, in the Toolbox, click the **Report** icon.
3. Outline the area under your graph where you want your report to be located on the page.

4. Drag **SALESREP** to the new report area and a Query palette opens. Drop **SALESREP** in the **By** folder. See Figure 7-19.

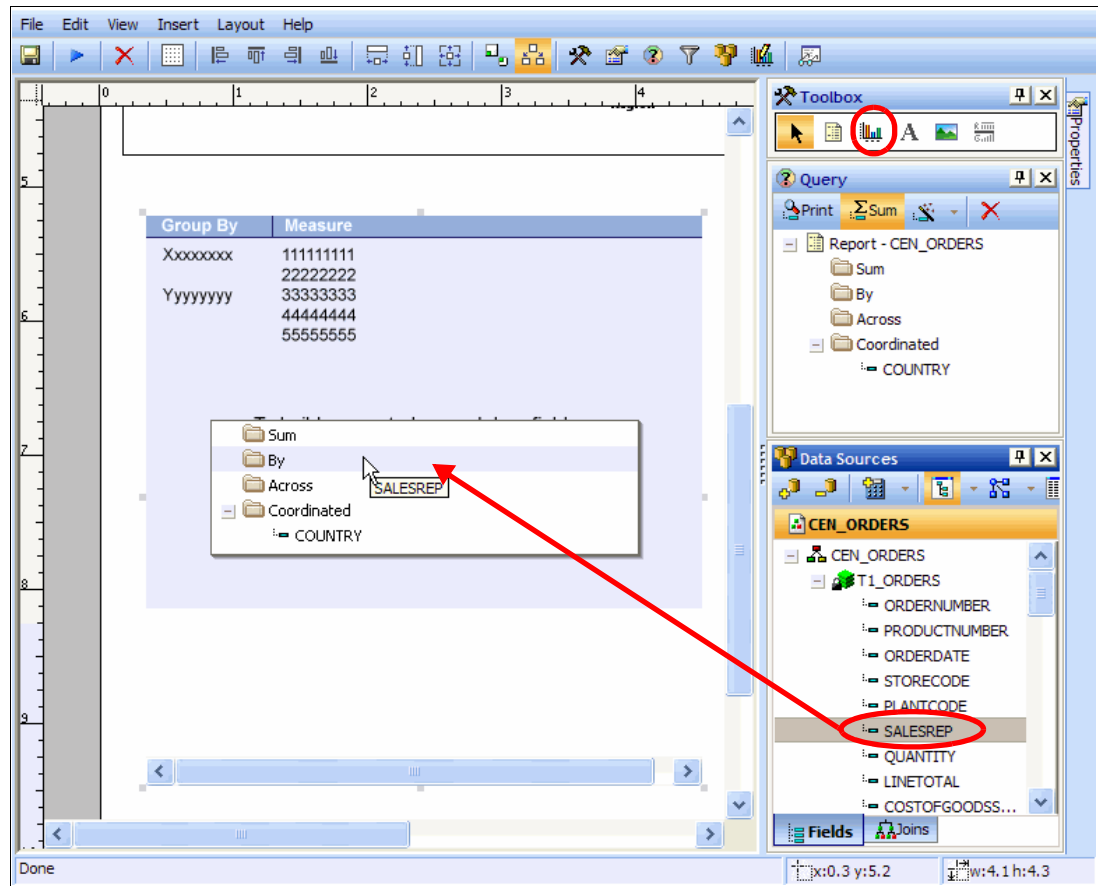


Figure 7-19 Adding fields to a report

You can drag fields directly to the report, or you can drag them to a query palette that you open and dock.

Tip: When creating a report, you can select one or multiple fields from the data source list and then drag them to the canvas. DB2 Web Query automatically creates a region for the report, autosizes it, and places alpha fields in the Sort By folder and numeric fields in the Sum folder. Try it. If you have a small table or view, you can drag the entire table to the canvas and have an instant one-click report.

5. Drag **ORDERNUMBER** into the **By** folder (Figure 7-20).

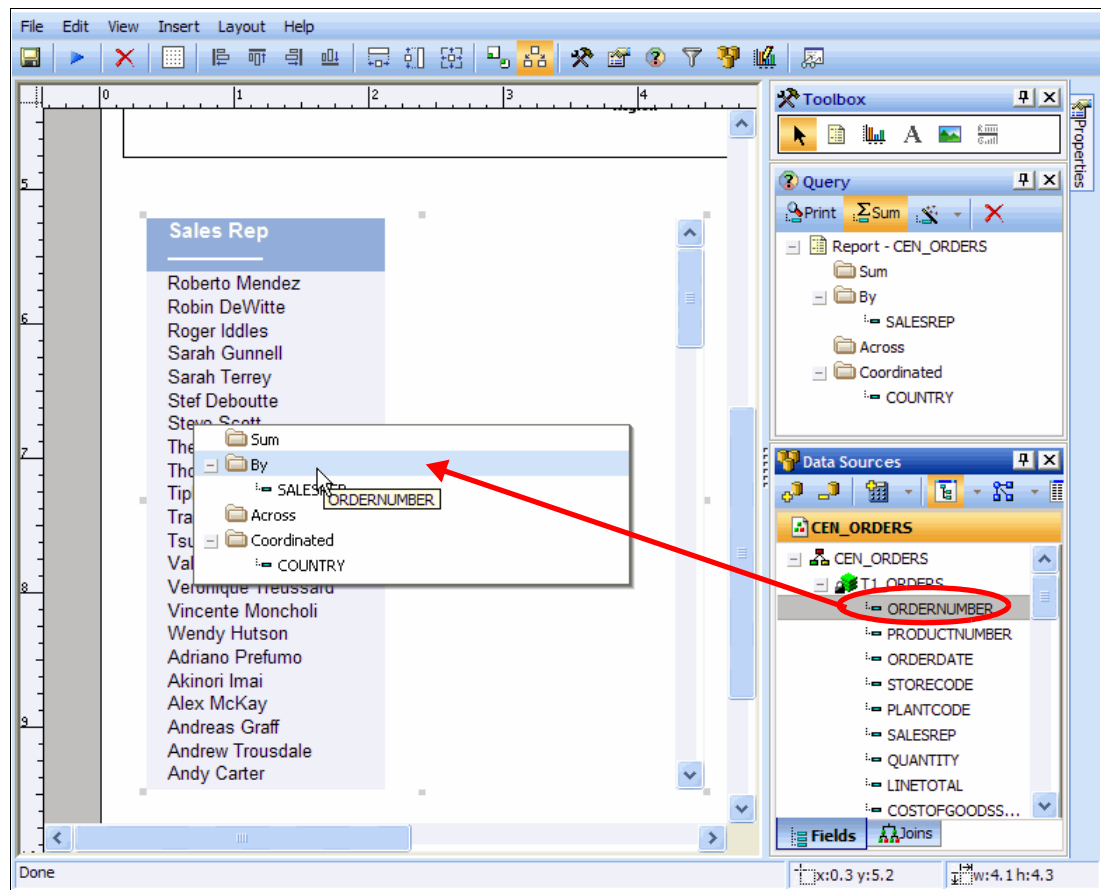


Figure 7-20 Adding additional fields to the report

6. Drag the remaining columns as shown in Figure 7-21 onto the Query palette. You do not have to drag COUNTRY under the Coordinated folder.

After you place a column under the Coordinated folder, it is automatically there for every report or graph in your layout.

You have now specified the columns to use in your query.

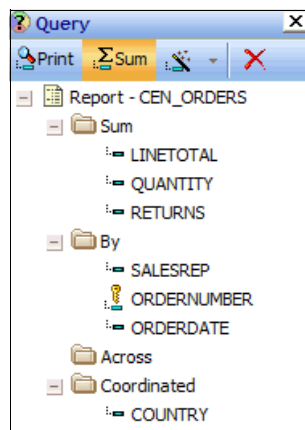


Figure 7-21 Completed query palette

Tip: If your data files are large and complex, consider changing the Data preview option under Document properties from Live data to Sample data. The Design record limit controls the number of live data records read while designing and previewing the report.

7. Add some subtotals when the sales representative changes and when the country changes. Also add grand totals at the end of the report:
 - a. Open the query palette if it is closed.
 - b. Highlight **SALESREP** and go to the Properties pane. Scroll down to **Subtotal** and click the **ellipsis (...)** button.
 - c. In the Sub Total Options window (Figure 7-22), select the **Sub Totals** radio button. In the Current Subtotal Text field, type `subtotal for` and click **OK**.

The default for a column is to sum or total it. This can be overridden, and you can request column functions such as average or count.

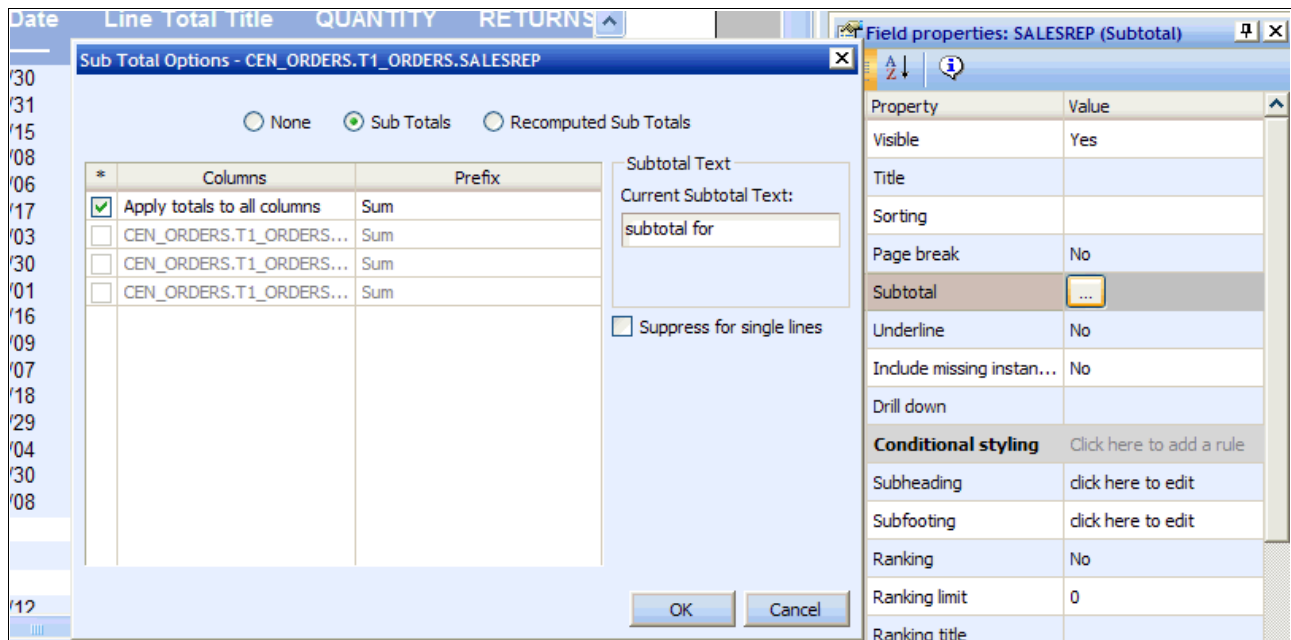


Figure 7-22 Adding subtotals

- d. Add the grand totals for each section or country in our report. To do this, we need Country as a By column not just a Coordinated column.
 - i. Select **Country** from the Data Sources pane, add it as a **By** column, and then move it to the top of the By column list.
 We do not want to see a column for Country in our report since it is the same on the entire page. We must hide the column, but still have the properties available to indicate that subtotalling is desired.
 - ii. Open the Query palette if it is closed.
 - iii. Highlight **COUNTRY** under the By folder and go to the **Properties** display.
 - iv. In the Field properties panel, for Visible, select **No**. Scroll down, highlight subtotal and click the **ellipsis** button.

- v. In the Sub Total Options window (Figure 7-23), click the **Sub Totals** radio button. Under Current Subtotal Text, type **Grand totals for** and click **OK**.

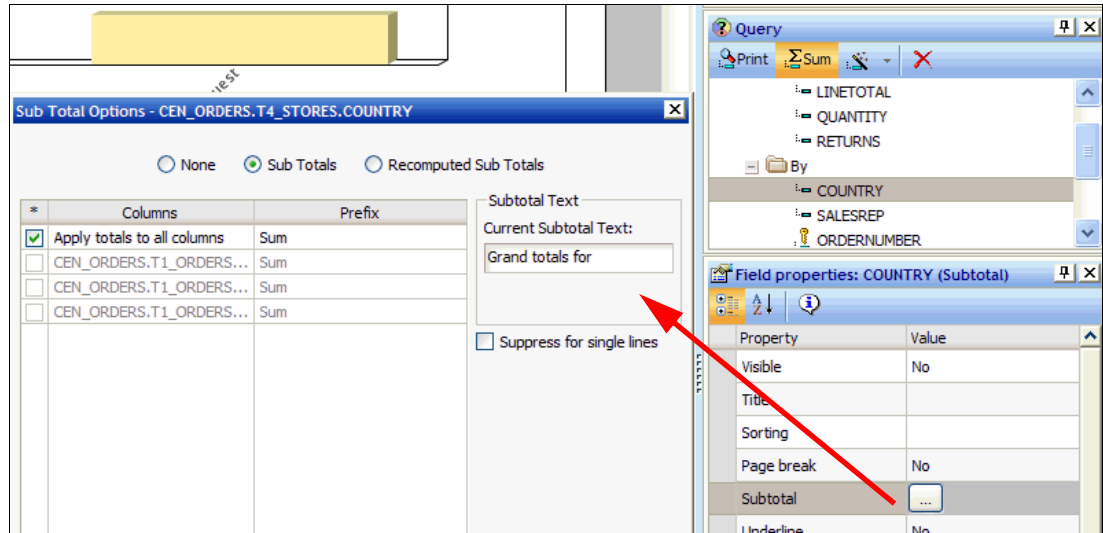


Figure 7-23 Grand totals for each country

- e. Add the grand totals to the report. These totals are not associated with any specific column changing but rather with the report itself.
 - i. Highlight the report and go to the **Report** properties panel (Figure 7-24).
 - ii. In the Report properties pane, locate Column totals and select **Totals**.

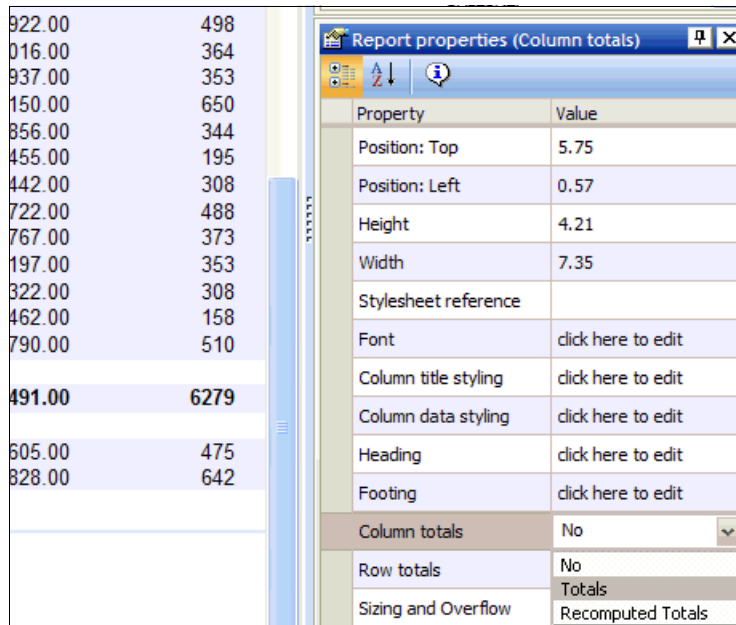


Figure 7-24 Column totals

8. If the report is longer than the frame that we have drawn, complete these steps:
 - a. Highlight the report so that you can work with the **Report** properties.
 - b. The Overflow property controls what happens when the report has too many rows to be displayed in the defined region. The default value of *Fixed* indicates that the report is restricted to the exact area that you defined. If it overflows, the rows continue in the same area on the next page until the report is complete. If you select *Flowing*, the report ignores the vertical boundaries and flows from the top to the bottom of each following page until it is complete.

We want our report to take full pages and flow until it is finished. Make sure that the Sizing and Overflow parameter is set to **Flowing** (Figure 7-25).

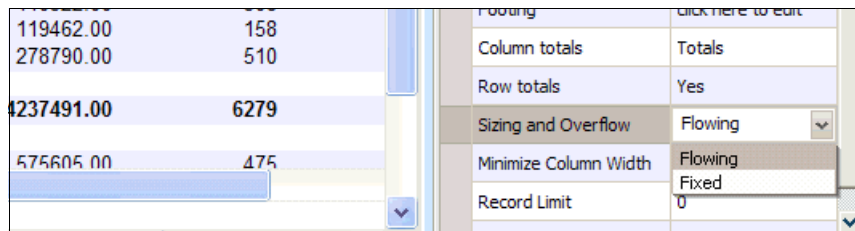


Figure 7-25 Controlling report overflow

9. Save and run your report. Figure 7-26 shows the results of running this report.

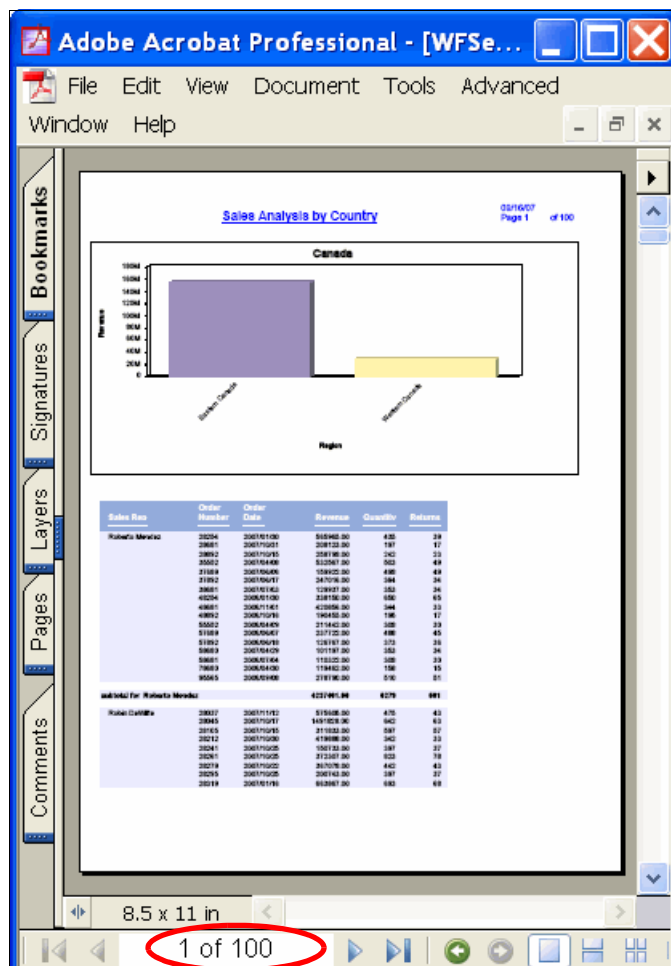


Figure 7-26 Page 2 of a flowing report

10. Notice that the report is quite long as indicated by the number of pages at the bottom of the window ("1 of 100"). Trying to find any individual country can be time consuming. To solve this problem, add a table of contents with an entry every time the COUNTRY changes; we have a page break and the report layout starts over again.

From the Query palette under Coordinated, highlight **COUNTRY**. In the Field properties pane, and for Table of contents, select **Yes** (Figure 7-27).

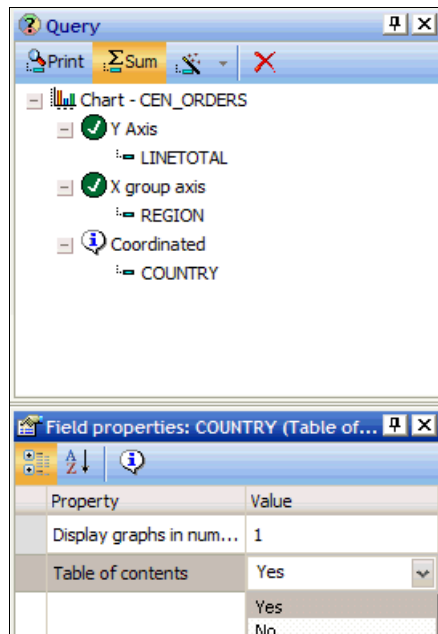


Figure 7-27 Table of contents property

11. Save and run your report. Figure 7-28 shows the results of running this report.

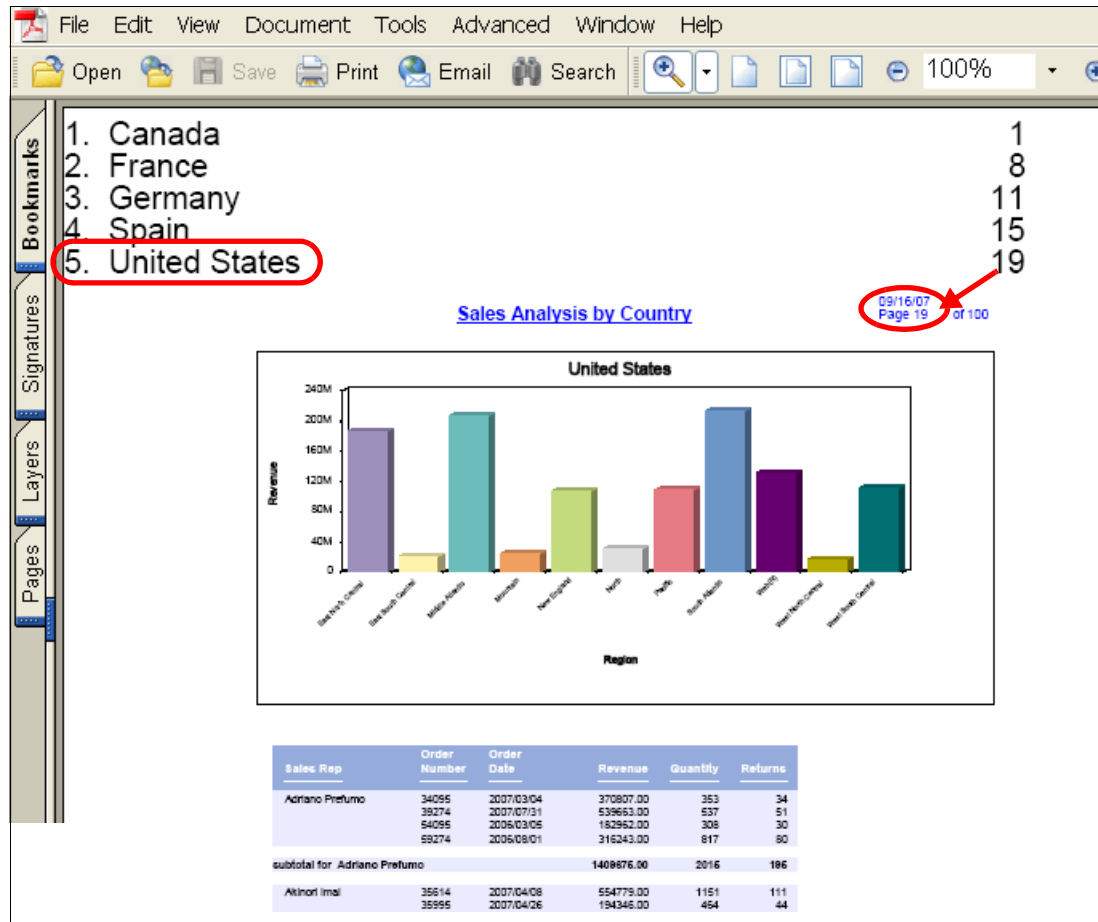


Figure 7-28 PDF with a table of contents

You have now created your first compound PDF report using Power Painter.

Note: There is a lot more that you can do in Power Painter. This tutorial has barely scratched the surface. Try dragging multiple fields at once on the page, and you will have a new report. You can also try looking at “relationships” and using the advanced graphics. There is even a speedometer, which is helpful for dashboards.

You can open any Report Assistant or Graph Assistant report in Power Painter. This gives you a starting point before you continue to add new reports using the Power Painter functionality.



Part 3

Tutorials for DB2 Web Query optional features

In Part 2, “Tutorials for DB2 Web Query” on page 55, we cover the base features that are included in DB2 Web Query. In this part, we cover the optional components that are ordered as feature codes for 5733-QU2.

This part includes the following chapters:

- ▶ Chapter 8, “Active Reports” on page 175
Active Reports enable the analysis of reports by users even when they are not connected to the System i platform.
- ▶ Chapter 9, “Developer Workbench” on page 193
This chapter is a prerequisite for both Chapters 10 and 11.
- ▶ Chapter 10, “Online analytical processing” on page 201
The graphical hierarchy or dimension design components of Developer Workbench are a prerequisite for OLAP-enabling your reports. After you define these hierarchical structures, it is simply a matter of the report developer including them in the existing reports.
- ▶ Chapter 11, “HTML Layout Painter” on page 233
HTML Layout Painter is a component of Developer Workbench. With it, you can create compound reports, dashboards, and highly parameterized reports.
- ▶ Chapter 12, “Additional features of Developer Workbench” on page 265
Developer Workbench is a set of advanced tools for programmers. This chapter includes information about advanced metadata definition, data profiling, impact analysis, SQL wizards, and business view creation of your data.



Active Reports

Active Reports is an optional feature that is available from IBM for DB2 Web Query. While the base product is licensed to named users, Active Reports is licensed to a System i machine, based strictly on the software tier (P10, P20, and so on). Any user who can create a report using either Report Assistant, Graph Assistant, or Power Painter is entitled to create an Active Report. Since Active Reports is intended for disconnected users, individual users do not require a license to use the reports.

The HTML reports must be made available to users either by e-mail, a shared network drive, or additional tools, such as Report Caster, from Information Builders. As you will see in this chapter, with Active Reports, the data and the reporting controls are all stored within the HTML page. When the user has access to the HTML page, the user can access, manipulate, and analyze the data with no requirement for any additional licensing.

With Active Reports, for example, a sales person might sign on to the System i environment in the morning and run a report for each customer they plan to visit that day. Then they can save the reports on their local hard drive. From their car, the sales person can then look at and analyze the data for their customer immediately prior to visiting them.

Attention: To use Active Reports, the server requires a license key that can be purchased as an optional feature of DB2 Web Query (5733-QU2).

8.1 Overview and highlights of Active Reports

Active Reports allows users to interact with their reports and work disconnected from the System i environment. HTML Active Report pages are self-contained reports. Both the data and the Java Script are compressed within a single efficient file. This file can be stored by the users on their local drive. It can be stored on the server for retrieval by the users or it can be sent by e-mail to users.

Active Reports are well suited for the mobile worker who is frequently disconnected from the Web. With Active Reports, you can create a variety of simple or advanced charts (pie, line, bar, or scatter), roll-up analysis, or pivot reports. You might want to allow your sales people to analyze the buying patterns for the clients they intend to visit that day. Your service personnel might want to view the service history of a customer before they visit. You might even choose to send reports by e-mail to your customers or partners that detail their activities with you. The user of an Active Report requires no special software, and there are no licensing fees for users.

With Active Reports, users can perform the following tasks:

- ▶ Create graphs and charts based on their data
- ▶ Filter or highlight their data
- ▶ Change the report sort sequence to any column
- ▶ Apply calculations such as sum, min, max, avg, count and count distinct to their data
- ▶ Change the look of a report by hiding columns, changing the pagination, and more
- ▶ Add data visualization bars to compare and contrast values in a column

Users can continue to work with their reports in a browser window, download the reports to Microsoft Excel, or even pull their data and graphs directly into Microsoft PowerPoint®. Figure 8-1 on page 177 shows a sample of the output generated from Active Reports.

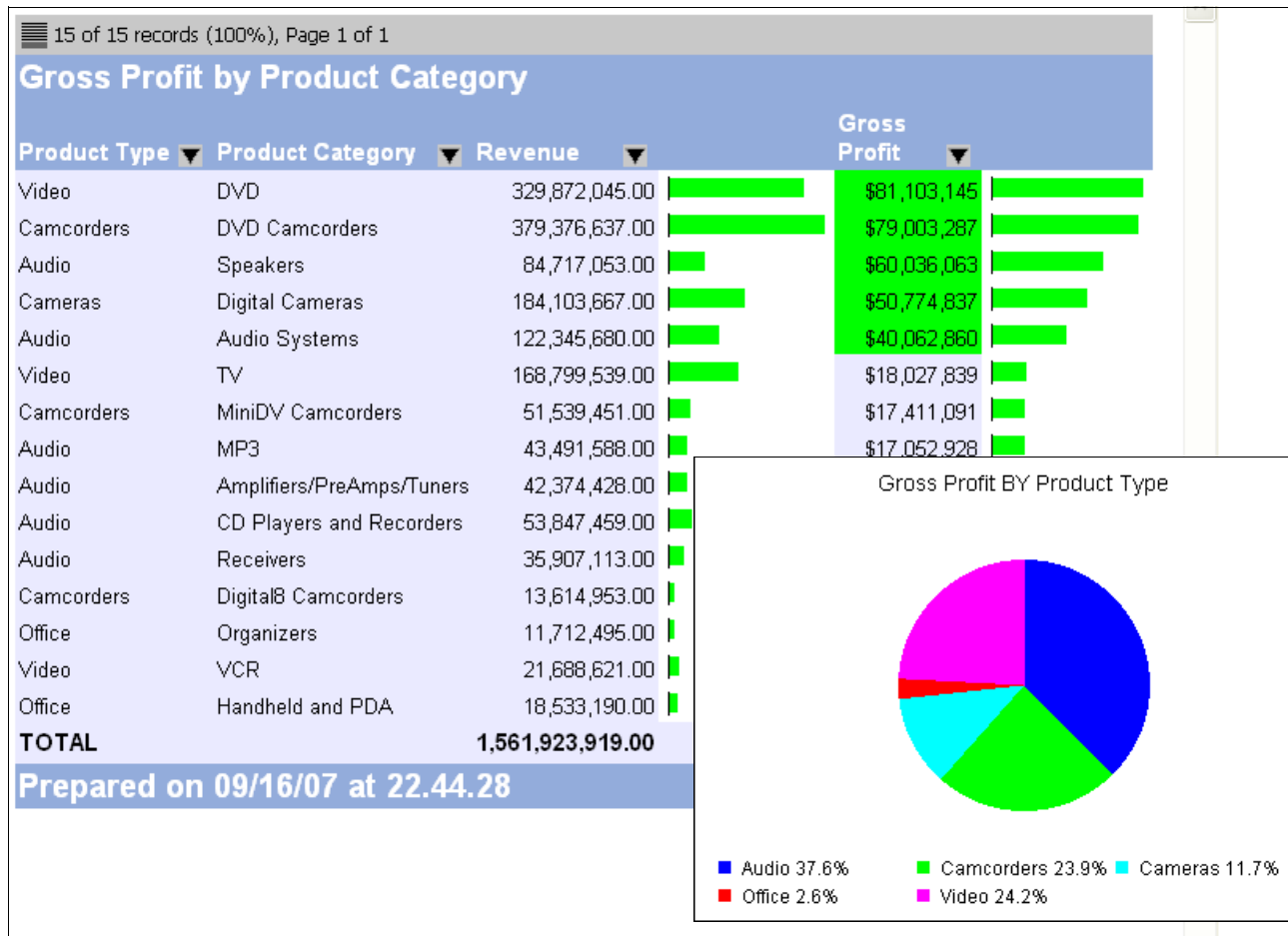


Figure 8-1 Active Reports sample

8.2 Creating a report with Active Report

We enable the RA3_GrossPft report that we created in Chapter 5, “Report Assistant” on page 63, to be used when the user is not connected to the server.

Important: Prior to beginning this tutorial, you *must* have upgraded your license key. The only requirement for Active Reports is that the server be properly licensed. There are no user requirements.

1. From the DB2 Web Query home page, open the **RA3_GrossPft** report.
2. Click the **Report options** tab.

3. On the Report options tab (Figure 8-2), complete these steps:
 - a. For Generate report as, select **AHTML - Active Report**. When you select this option, notice that the Active Report Styling button becomes active.

At this point, you might run the report. However we a make a few formatting and highlighting changes first.

Note: Horizontal (Across) sort fields are not supported for HTML Active Reports.

- b. Click the **Active Report Styling** button.

The screenshot shows the 'Report options' tab of a software application. The 'Generate report as' dropdown is set to 'AHTML - Active Report'. The 'Output format spec' dropdown is also set to 'AHTML - Active Report'. The 'Report title' is 'AHTML - Active Report'. The 'Interactive options' section has 'Enable OLAP' set to 'Off' and 'Automatic drill down' set to 'Off'. The 'Global report styling' section has 'Active Report Styling' button active, 'Apply an existing WebFOCUS Style Sheet' set to 'ibm_blue1', and 'Apply an existing Cascading Style Sheet' set to 'NONE'. The 'Database Output' section has 'Destination/Library' set to 'HOLD' and 'Overwrite existing file' checked.

Figure 8-2 Selecting the Active Reports option

- c. In the Active Report Styling window (Figure 8-3), under Report view, select **Tabular**. For Row selection, select **yellow**. For Visualization colors, select **red** for Negative and **dark green** for Positive. Click **OK**.

Note: In this window, we define the presentation and initial look of the report for the user. We discuss Accordion reports later in the tutorial.

The dialog box is titled "Active Report Styling -- Web Page Dialog". It contains several sections for configuring the report's appearance:

- Report view:** Radio buttons for "Tabular" (selected) and "Accordion".
- Freeze columns:** A dropdown menu set to "None".
- Pagination options:**
 - Justification: A dropdown menu set to "Center".
 - Records per page: A dropdown menu set to "20".
 - Text: A color selection button.
 - Background: A color selection button.
- Row selection colors:**
 - Hover: A color selection button showing yellow.
 - Selected: A color selection button showing cyan.
- Visualization colors:**
 - Negative: A color selection button showing red.
 - Positive: A color selection button showing dark green.
- Calculations:**
 - Colors: Two color selection buttons for "Values" and "Background".
 - Location: A dropdown menu set to "Top row".
- Menu options:**
 - Font color: Two color selection buttons for "Text" and "Hover".
 - Menu color: Three color selection buttons for "Back", "Hover", and "Border".

At the bottom are "OK" and "Cancel" buttons.

Figure 8-3 Active Report Styling

4. Save your report as AR1_GrossPft.
5. Run your report. Figure 8-4 shows the results of running the report.

15 of 15 records (100%), Page 1 of 1

Gross Profit for Type Audio				
Product Type ▼	Product Category ▼	Revenue ▼	Cost of Goods Sold ▼	Gross Profit ▼
Audio	Speakers	84,717,053.00	24,680,990.00	\$60,036,063
	Audio Systems	122,345,680.00	82,282,820.00	\$40,062,860
	MP3	43,491,588.00	26,438,660.00	\$17,052,928
	Amplifiers/PreAmps/Tuners	42,374,428.00	25,739,570.00	\$16,634,858
	CD Players and Recorders	53,847,459.00	37,838,460.00	\$16,008,999
	Receivers	35,907,113.00	22,998,000.00	\$12,909,113
Subtotal for Audio		382,683,321.00	219,978,500.00	\$162,704,821
Camcorders	DVD Camcorders	379,376,637.00	300,373,350.00	\$79,003,287
	MiniDV Camcorders	51,539,451.00	34,128,360.00	\$17,411,091
	Digital8 Camcorders	13,614,953.00	6,512,600.00	\$7,102,353
Subtotal for Camcorders		444,531,041.00	341,014,310.00	\$103,516,731

Figure 8-4 Active Reports in the tabular format

- Click a down arrow next to any column and select **Save Changes** (Figure 8-5). Save your report as AR1_GrossPft.html.

Note: You can also use the Save As option in your browser to save the HTML page. If you choose the Export option instead of the Save Changes option, you export the records without formatting and controls. The records can be exported to an HTML page, Excel, and comma separated values (CSV) files.

15 of 15 records (100%), Page 1 of 1

Gross Profit for Type Audio

Product Type ▼	Product Category ▼	Revenue ▼	Cost of Goods Sold ▼	Gross Profit ▼
Audio	Speakers	Sort Ascending	24,680,990.00	\$60,036,063
	Audio Systems	Sort Descending	82,282,820.00	\$40,062,860
	MP3		26,438,660.00	\$17,052,928
	Amplifiers/PreAmps/Tune	Filter ▶	25,739,570.00	\$16,634,858
	CD Players and Recorde	Calculate ▶	37,838,460.00	\$16,008,999
	Receivers	Chart ▶	22,998,000.00	\$12,909,113
	Subtotal for Audio	Rollup ▶	219,978,500.00	\$162,704,821
Camcorders	DVD Camcorders	Pivot (Cross Tab) ▶	300,373,350.00	\$79,003,287
	MiniDV Camcorders		34,128,360.00	\$17,411,091
	Digital8 Camcorders	Hide Column	6,512,600.00	\$7,102,353
	Subtotal for Camcorders	Unhide Columns ▶	341,014,310.00	\$103,516,731
Cameras	Digital Cameras	Freeze Column	133,328,830.00	\$50,774,837
	Subtotal for Cameras	Unfreeze All	133,328,830.00	\$50,774,837
Office	Organizers		4,957,305.00	\$6,755,190
	Handheld and PDA	Show Records ▶	14,067,420.00	\$4,465,770
	Subtotal for Office	Send as E-mail	19,024,725.00	\$11,220,960
Video	DVD	Save Changes	248,768,900.00	\$81,103,145
	TV	Export ▶	150,771,700.00	\$18,027,839
	VCR	Window ▶	16,270,950.00	\$5,417,671
	Subtotal for Video	Restore Original	415,811,550.00	\$104,548,655
	TOTAL		1,129,157,915.00	\$432,766,004

Figure 8-5 Saving your Active Report for offline use

Tip: After you save your report as an HTML file, you can send it to users by using e-mail. In this case, the recipient can view and analyze the report with no connectivity to the System i environment. To run and save an Active Report, you must be a named user. To work with the HTML file after it is created, you do not need to be a named user.

In the next section, we continue to work with this Active Report, but we do this so as though we are the disconnected user and not the developer.

8.3 Using an Active Report

When a user receives an Active Report, they open it in their browser. After your users receive this report, they can then perform various types of analysis on it without connecting to the System i environment.

ActiveX® controls: Active Reports use ActiveX controls. Based on your browser security settings, you might see warning messages, and you might be asked if you want to allow blocked content.

1. Locate and open the **AR1_GrossPft.html** report. If you are connected to the System i environment, you can also go to the DB2 Web Query home page, navigate to the report, and run the report.
2. Analyze the product categories and evaluate the gross profit compared with revenue or Line Total. Click the down arrow next to **Gross Profit** and select **Sort Descending** (Figure 8-6).

Product Type ▼	Product Category ▼	Revenue ▼	Cost of Goods Sold ▼	Gross Profit ▼	
Audio	Amplifiers/PreAmps/Tuners	42,374,428.00	25,739,570.00	\$16,634,8	Sort Ascending
	Audio Systems	122,345,680.00	82,282,820.00	\$40,062,8	Sort Descending
	CD Players and Recorders	53,847,459.00	37,838,460.00	\$16,008,9	Filter
	MP3	43,491,588.00	26,438,660.00	\$17,052,9	Calculate
	Receivers	35,907,113.00	22,998,000.00	\$12,909,1	Chart
	Speakers	84,717,053.00	24,680,990.00	\$60,036,0	

Figure 8-6 Active Reports Sort Descending

3. To make the report fit better on the page, we hide the Cost of Goods Sold column. To hide a column, click the down arrow next to **Cost of Goods Sold** and choose **Hide Column** (Figure 8-7).

Product Type ▼	Product Category ▼	Revenue ▼	Cost of Goods Sold ▼	Gross Profit ▼	
Video	DVD	329,872,045.00	248,768,900.		Sort Ascending
Camcorders	DVD Camcorders	379,376,637.00	300,373,350.		Sort Descending
Audio	Speakers	84,717,053.00	24,680,990.		Filter
Cameras	Digital Cameras	184,103,667.00	133,328,830.		Calculate
Audio	Audio Systems	122,345,680.00	82,282,820.		Chart
Video	TV	168,799,539.00	150,771,700.		Rollup
Camcorders	MiniDV Camcorders	51,539,451.00	34,128,360.		Pivot (Cross Tab)
Audio	MP3	43,491,588.00	26,438,660.		Visualize
Audio	Amplifiers/PreAmps/Tuners	42,374,428.00	25,739,570.		Hide Column
Audio	CD Players and Recorders	53,847,459.00	37,838,460.		Unhide
Audio	Receivers	35,907,113.00	22,998,000.		

Figure 8-7 Hiding columns

- The Visualize feature is a powerful tool that allows you to see, at a glance, relationships between various columns. Click the down arrow for the **Gross Profit** column and select **Visualize**. See Figure 8-8.

Product Type ▼	Product Category ▼	Revenue ▼	Gross Profit ▼	
Video	DVD	329,872,045.00	\$81,103,145	Sort Ascending Sort Descending
Camcorders	DVD Camcorders	379,376,637.00	\$79,003,287	Filter ▶
Audio	Speakers	84,717,053.00	\$60,036,063	Calculate ▶
Cameras	Digital Cameras	184,103,667.00	\$50,774,837	Chart ▶
Audio	Audio Systems	122,345,680.00	\$40,062,860	Rollup ▶
Video	TV	168,799,539.00	\$18,027,839	Pivot (Cross Tab) ▶
Camcorders	MiniDV Camcorders	51,539,451.00	\$17,411,091	Visualize
Audio	MP3	43,491,588.00	\$17,052,928	
Audio	Amplifiers/PreAmps/Tuners	42,374,428.00	\$16,634,858	
Audio	CD Players and Recorders	53,847,459.00	\$16,008,999	

Figure 8-8 Active Reports Visualize Selection

Notice the green or red bars that are displayed in the chart in Figure 8-9. Visualize gives you a quick visual representation of the data in a column.

- Click the down arrow for the **Revenue** column and select **Visualize**.

Now you see the bars for the Revenue column as well. With the visualize bars, it is much easier to spot relationships. Notice that although speakers is not ranked high in the Revenue column, they are extremely profitable and generate the third highest profit of any product category.

15 of 15 records (100%), Page 1 of 1				
Gross Profit by Product Category				
Product Type ▼	Product Category ▼	Revenue ▼		Gross Profit ▼
Video	DVD	329,872,045.00		\$81,103,145
Camcorders	DVD Camcorders	379,376,637.00		\$79,003,287
Audio	Speakers	84,717,053.00		\$60,036,063
Cameras	Digital Cameras	184,103,667.00		\$50,774,837
Audio	Audio Systems	122,345,680.00		\$40,062,860
Video	TV	168,799,539.00		\$18,027,839
Camcorders	MiniDV Camcorders	51,539,451.00		\$17,411,091
Audio	MP3	43,491,588.00		\$17,052,928
Audio	Amplifiers/PreAmps/Tuners	42,374,428.00		\$16,634,858
Audio	CD Players and Recorders	53,847,459.00		\$16,008,999
Audio	Receivers	35,907,113.00		\$12,909,113
Camcorders	Digital8 Camcorders	13,614,953.00		\$7,102,353
Office	Organizers	11,712,495.00		\$6,755,190
Video	VCR	21,688,621.00		\$5,417,671
Office	Handheld and PDA	18,533,190.00		\$4,465,770
TOTAL		1,561,923,919.00		\$432,766,004
Prepared on 09/16/07 at 22.44.28				

Figure 8-9 Active Reports Visualize results

6. Click the down arrow for **Gross Profit**. Select **Calculate** → **% of Total** (Figure 8-10).

15 of 15 records (100%), Page 1 of 1

Gross Profit by Product Category

Product Type ▼	Product Category ▼	Revenue ▼		Gross Profit ▼	
Video	DVD	329,872,045.00		\$81,103,1	
Camcorders	DVD Camcorders	379,376,637.00		\$79,003,2	
Audio	Speakers	84,717,053.00		\$60,036,0	
Cameras	Digital Cameras	184,103,667.00		\$50,774,8	
Audio	Audio Systems	122,345,680.00		\$40,062,8	
Video	TV	168,799,539.00			
Camcorders	MiniDV Camcorders	51,539,451.00			
Audio	MP3	43,491,588.00			
Audio	Amplifiers/PreAmps/Tuners	42,374,428.00			
Audio	CD Players and Recorders	53,847,459.00			
Audio	Receivers	35,907,113.00			
Camcorders	Digital8 Camcorders	13,614,953.00			
Office	Organizers	11,712,495.00			
Video	VCR	21,688,621.00			
Office	Handheld and PDA	18,533,190.00		\$4,465,0	
TOTAL		1,561,923,919.00		\$432,766,0	

Prepared on 09/16/07 at 22.44.28

Sort
Ascending
Sort
Descending
Filter
Calculate
Chart
Rollup
Pivot (Cross
Tab)
Visualize
Hide Column
Unhide
Columns
Freeze
Column
Unfreeze All
Show Records
Send as E-
mail
Save Changes
Export
Window
Restore
Original

Clear
Clear All
Sum
Avg
Min
Max
Count
Distinct
Detail
% of Total

Figure 8-10 Adding a % of Total column

Notice that a new column, called % of Total, is now displayed that shows each product category row as a percent of the total Gross Profit in this report (Figure 8-11).

- Click the down arrow for **Gross Profit** and select **Sum**. Your summary prints at the top of the page by default.

15 of 15 records (100%), Page 1 of 1

Gross Profit by Product Category

Product Type ▼	Product Category ▼	Revenue ▼		Gross Profit ▼	% of Total
				Total Sum \$432,766,004	
Video	DVD	329,872,045.00		\$81,103,145	
Camcorders	DVD Camcorders	379,376,637.00		\$79,003,287	
Audio	Speakers	84,717,053.00		\$60,036,063	
Cameras	Digital Cameras	184,103,667.00		\$50,774,837	
Audio	Audio Systems	122,345,680.00		\$40,062,860	
Video	TV	168,799,539.00		\$18,027,839	4.17%
Camcorders	MiniDV Camcorders	51,539,451.00		\$17,411,091	4.02%
Audio	MP3	43,491,588.00		\$17,052,928	3.94%
Audio	Amplifiers/PreAmps/Tuners	42,374,428.00		\$16,634,858	3.84%
Audio	CD Players and Recorders	53,847,459.00		\$16,008,999	3.70%
Audio	Receivers	35,907,113.00		\$12,909,113	2.98%
Camcorders	Digital8 Camcorders	13,614,953.00		\$7,102,353	1.64%
Office	Organizers	11,712,495.00		\$6,755,190	1.56%
Video	VCR	21,688,621.00		\$5,417,671	1.25%
Office	Handheld and PDA	18,533,190.00		\$4,465,770	1.03%
TOTAL		1,561,923,919.00		\$432,766,004	

Prepared on 09/16/07 at 22.44.28

Figure 8-11 New % of Total column and Total Sum

- Filter the report temporarily so that we can concentrate on our audio sales. Click the down arrow for **Product Type** and choose **Filter** → **Equals** (Figure 8-12).

15 of 15 records (100%), Page 1 of 1

Gross Profit by Product Category

Product Type	Product Category	Revenue	Gross Profit	% of Total
Total Sum \$432,766,004				
Video		329,872,045.00	\$81,103,145	
Camcorders		379,376,637.00	\$79,003,287	
Audio		84,717,053.00	\$60,036,063	
Cameras			\$50,774,837	
Audio			\$40,062,860	
Video			\$18,027,839	4.17%
Camcorders			\$17,411,091	4.02%
Audio			\$17,052,928	3.94%
Audio			\$16,634,858	3.84%
Audio			\$16,008,999	3.70%
Audio			\$12,909,113	2.98%
Camcorders			\$7,102,353	1.64%
Office			\$6,755,190	1.56%
Video			\$5,417,671	1.25%
Office			\$4,465,770	1.03%
TOTAL		1,561,923,919.00	\$432,766,004	
Prepared on		12.44.28		

Figure 8-12 Active Reports filtering

- In the Filter Selection window (Figure 8-13), click the down arrow on the right to see a list of the distinct values in a field. Select **Audio** from the list. Then click **Filter**.

Highlight button: If you click the Highlight button instead of clicking the Filter button, your report contains all the records, but highlights the ones that meet the filter criteria.

Filter Selection

Product Type Equals

Operator: AND Add Condition Filter HighLight Clear All

1 Organizers 11,712,495.00

2 VCD 21,688,631.00

3 3,360.00

4 1,660.00

Audio

Camcorders

Cameras

Office

Video

Figure 8-13 Active Reports Where Equals clause

Contains operand: Besides your standard operators, such as Greater Than and Equal, you can search for a string of data anywhere within a field using the Contains operand. By default, Contains is not case sensitive. If case is important, there is an additional operand Contains (match case). There are also similar Omits and Omits (match case) options.

10. In the filtered chart (Figure 8-14), you see both your filtered sum and the total sum for a column. In the upper left corner, click the new **SUB/TOT** icon. It enable you to cycle through viewing the sum of all your data, the sum of your filtered data, and having both sums displayed concurrently.

At any point, you can choose the Restore Original option. To see all the records, close the Filter Selection window.

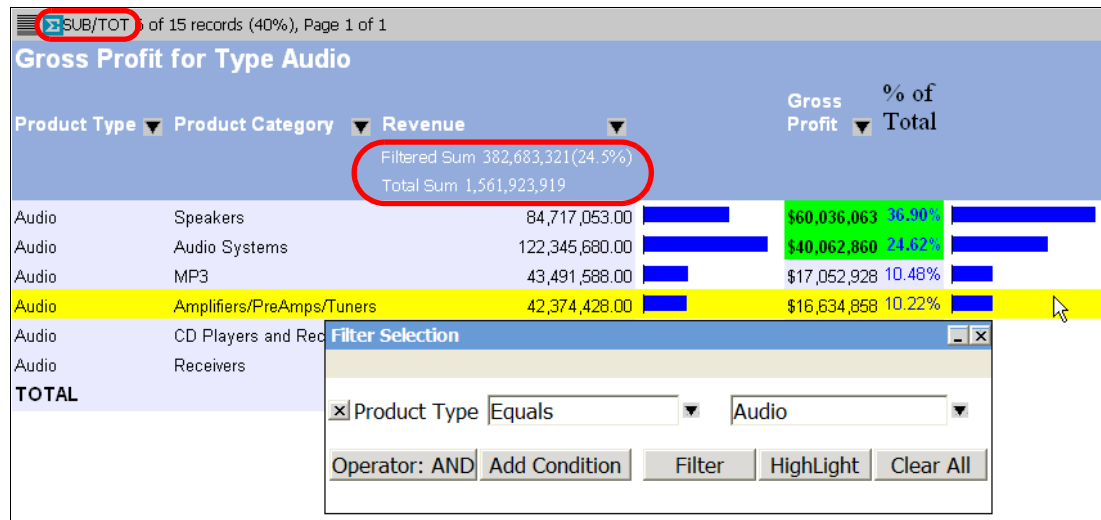


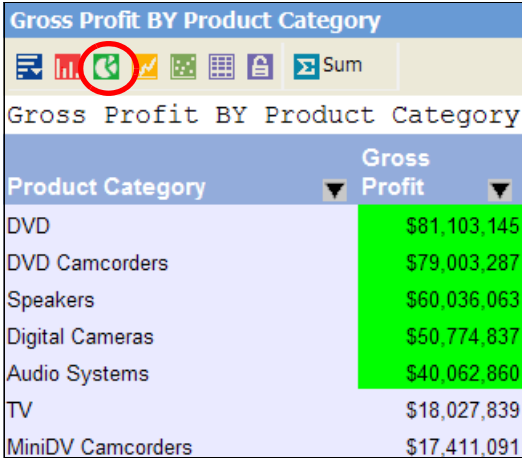
Figure 8-14 Active Reports Filtered/Total Sum

11. A helpful way to analyze the data involves changing the grouping and looking at your numbers slightly differently. Select the down arrow for **Gross Profit** and select **Rollup** → **Group By Product Category** (Figure 8-15).



Figure 8-15 Active Reports Rollup grouped by Product Category

12. Click the **Pie chart graph** icon to change the report that you are looking at into a pie chart (Figure 8-16). You can choose line or bar charts as well.



Product Category	Gross Profit
DVD	\$81,103,145
DVD Camcorders	\$79,003,287
Speakers	\$60,036,063
Digital Cameras	\$50,774,837
Audio Systems	\$40,062,860
TV	\$18,027,839
MiniDV Camcorders	\$17,411,091

Figure 8-16 Active Report rollup results

13. Export the chart to Microsoft PowerPoint, Excel, and Word. Click the first **List type** icon and select **Export to** → **Excel** (Figure 8-17).

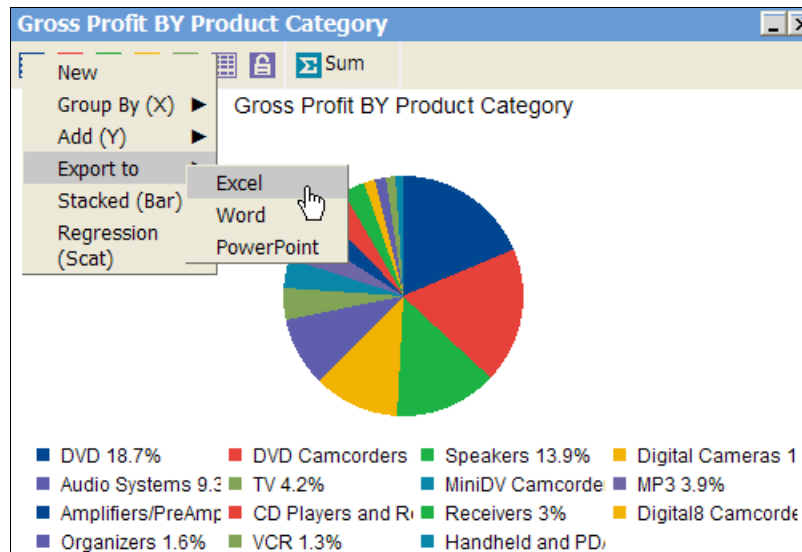


Figure 8-17 Active Reports graph

You can access the same graph directly from your full report by clicking the Gross Profit column, selecting Chart, Pie, and Product Category as your Group by field.

Tip: If you created filters, then plotted your graph, and closed the filter, by default, your graph changes to reflect the displayed report or all the data. To keep the chart as is, displaying the filtered data, select the **Lock** icon.

Figure 8-18 shows the output of the report to Excel.

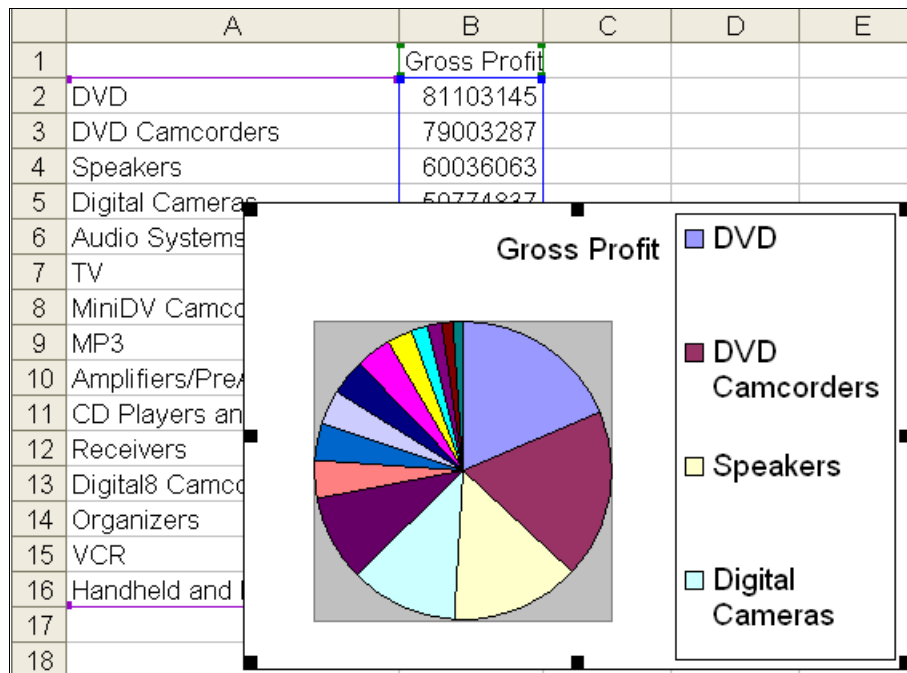


Figure 8-18 Active Reports output to Excel

14. Export to the other Microsoft product if you have them loaded on your system.
15. Close Excel and close the charting window.
16. To use the Cross tab or Pivot functionality, select the down arrow for **Gross Profit** and select **Pivot** → **Product Category** → **Product Type** as the field to go across the top of your table (Figure 8-19).

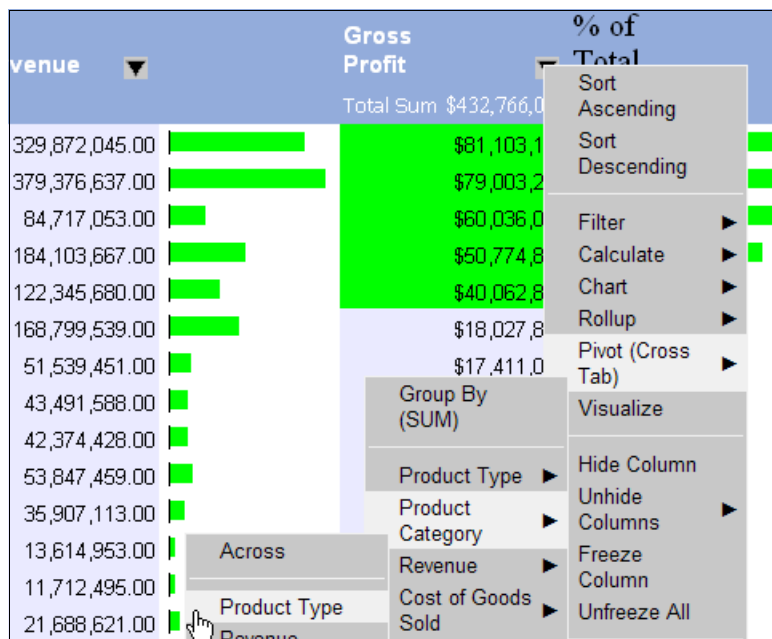


Figure 8-19 Creating pivot tables with Active Report

Note: This report is not a good one to demonstrate cross tab or pivot functionality. There are not enough fields shown that relate to each other, for us to pivot on. The next steps illustrate additional capabilities of Active Reports.

17. When you have a basic pivot table, you can start adding to it, changing it, and pivoting it. Click the left (circled in Figure 8-20) icon above Product Type.

Gross Profit BY Product Type, Product Category						
Sum						
Gross Profit BY Product Type, Product Category						
Product Type						
Product Category						
Audio	Camcorders	Cameras	Office	Video	Total	
Amplifiers/PreAmps/Tuners	\$16,634,858				\$16,634,858	
Audio Systems	\$40,062,860				\$40,062,860	
CD Players and Recorders	\$16,008,999				\$16,008,999	

Figure 8-20 Pivot Table output with Active Report

Clicking the icon to the left of the arrow moves a field back and forth between being a column or a row, pivoting on that field. Notice in Figure 8-21 that the title changes as the column layout changes.

Gross Profit BY Product Type, Product Category			
Product Type	Product Category		
Audio	Amplifiers/PreAmps/Tuners	\$16,634,858	
	Audio Systems	\$40,062,860	
	CD Players and Recorders	\$16,008,999	
	DVD		
	DVD Camcorders		
	Digital Cameras		
	Digital8 Camcorders		
	Handheld and PDA		
	MP3	\$17,052,928	
	MiniDV Camcorders		
	Organizers		
	Receivers	\$12,909,113	
	Speakers	\$60,036,063	

Gross Profit BY Product Category, Product Type			
Product Category	Product Type		
Amplifiers/PreAmps/Tuner	Audio	\$16,634,858	
	Camcorders		
	Cameras		
	Office		
	Video		
Audio Systems	Audio	\$40,062,860	
	Camcorders		
	Cameras		
	Office		
	Video		
CD Players and Recorders	Audio	\$16,008,999	
	Camcorders		

Figure 8-21 Changing sort breaks and sort priorities for Active Reports

There is quite a bit more that you can do with pivot tables if it makes sense for your data. You might ask to see the average profit numbers, for example, if you have more than one record per row grouped together.

Tip: To size the window shown in Figure 8-21 on page 189, scroll down to the bottom of the window. In the right corner is an area that you can drag to widen and size the pivot table. Click once outside the box after you finish dragging to release the cursor.

As you can see, Active Reports allow users to quickly add a level of analysis to their reports, all without being connected to the System i environment.

Let us look at the difference between a tabular and accordion Active Report:

1. If your report definition is closed, go to the DB2 Web Query home page navigate to it, right-click, and open it.
2. Click the **Field selection** tab (Figure 8-22).
3. On the Field selection tab, complete these steps:
 - a. Remove subtotals from PRODUCTTYPE.
 - b. Clear the **Add grand totals to the end of the report** check box.

Restriction: With earlier versions of the software, subtotal and total messages occasionally appeared in an accordion report. It is easiest to remove them now.

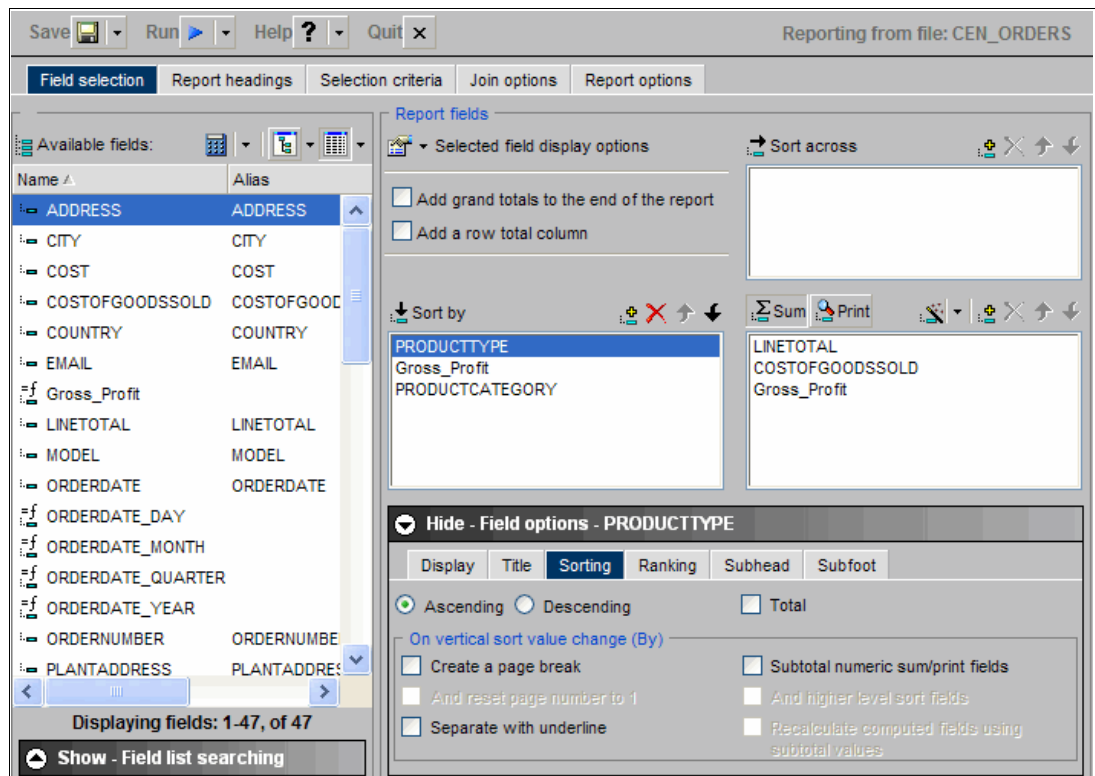


Figure 8-22 Removing subtotals and selecting the option to add grand totals to the end of the report

4. Click the **Report options** tab.
5. On the Report options tab, complete these steps:
 - a. Click the **Active Report Styling** button.
 - b. Select **Accordion** instead of Tabular and click **OK**.

- Run your report. Figure 8-23 shows the results of running the report with the Accordion option selected. Only the data values at the highest level (first vertical sort field) are shown initially.

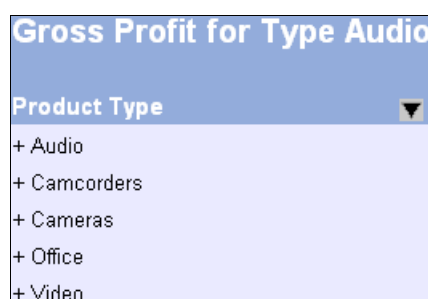


Figure 8-23 Accordion Active Report

- Click the plus sign (+) to expand a field. Click the minus sign (-) to compress the data. In our example, we expand **Audio** and see all the Product Categories within Audio (Figure 8-24). If we had more Sort By columns in our report, we could expand each of the levels.

15 of 15 records (100%), Page 1 of 1

Gross Profit for Type Audio				
Product Type ▼	Product Category ▼	Revenue ▼	Cost of Goods Sold ▼	Gross Profit ▼
- Audio				
	Speakers	84,717,053.00	24,680,990.00	\$60,036,063
	Audio Systems	122,345,680.00	82,282,820.00	\$40,062,860
	MP3	43,491,588.00	26,438,660.00	\$17,052,928
	Amplifiers/PreAmps/Tuners	42,374,428.00	25,739,570.00	\$16,634,858
	CD Players and Recorders	53,847,459.00	37,838,460.00	\$16,008,999
	Receivers	35,907,113.00	22,998,000.00	\$12,909,113
+ Camcorders				
+ Cameras				
+ Office				
+ Video				

Figure 8-24 Accordion format in Active Reports

- Look at some of the other options that are available under the arrows. You will see that you can export data to CSV files, to HTML files and directly into Excel. If you filter your data, you have the choice of exporting the entire data set or just the filtered records.

Tip: HTML Active Reports can support up to 20,000 rows of data. If you are using a Firefox browser, we recommend that you use only up to 15,000 rows of data. If you are using an Internet Explorer browser, use only 5,000 rows of data.

8.4 Summary

Active Reports is a tool that works extremely well when your users are remote and not connected to the System i environment. Even with local users, there are no named user licensing requirements to work with and analyze an Active Report. You can create an Active Report and send it by e-mail it to a user for the user to analyze it. The person who originally runs the report must be a licensed named user. The report is stored in a highly compressed format. Active Reports is just one more powerful reporting tool for you to use with DB2 Web Query.



Developer Workbench

Developer Workbench provides an integrated development environment (IDE) for building and maintaining a DB2 Web Query reporting environment. Developer Workbench includes additional tools to work with your metadata. The Synonym Editor allows you to change the way a field is displayed to the users. You can add commas and decimal signs, convert date formats, convert alpha to numeric fields, or even change the length of a field. The Synonym Editor also allows you to create calculated fields for users. You can predefine joins for the users or create business views that simplify the structure and number of columns the user sees. If you plan on using the optional online analytical processing (OLAP) feature, you must define your hierarchies or dimensions with Developer Workbench.

Using the metadata layer, Developer Workbench provides the ability to perform impact analysis at both a file level and a field level. If you must make changes to a field, you can see which reports and graphs use that field and the business views that it is part of.

Developer Workbench provides the capability to “profile” your data. Profiling allows you to see patterns in the text characters. For example, profiling telephone numbers in the Stores table might show us that five records have a format of nine numeric characters instead of ten. We might want to know if these five records are a quality problem. Data profiling also shows the minimum and maximum values in a column and identifies the outliers.

The SQL Wizard allows you to import SQL into Report or Graph Assistant.

One of the major tools within Developer Workbench is the HTML Layout Painter. HTML Layout Painter is specifically designed to create HTML pages that can contain compound reports or dashboards. You can find a complete tutorial for this component in Chapter 11, “HTML Layout Painter” on page 233.

In this chapter, we configure the System i DB2 Web Query environment and set up some Developer Workbench default settings. This chapter is a prerequisite for both Chapter 10, “Online analytical processing” on page 201, and Chapter 11, “HTML Layout Painter” on page 233. For more information about Developer Workbench, see Chapter 12, “Additional features of Developer Workbench” on page 265.

9.1 Configuring Developer Workbench

To configure Developer Workbench:

1. Because Developer Workbench is a PC application, install the software on the developer's PC.
2. When you start Developer Workbench the first time, add a definition for your System i environment.
 - a. Right-click **WebFOCUS** and select **Add** (Figure 9-1).

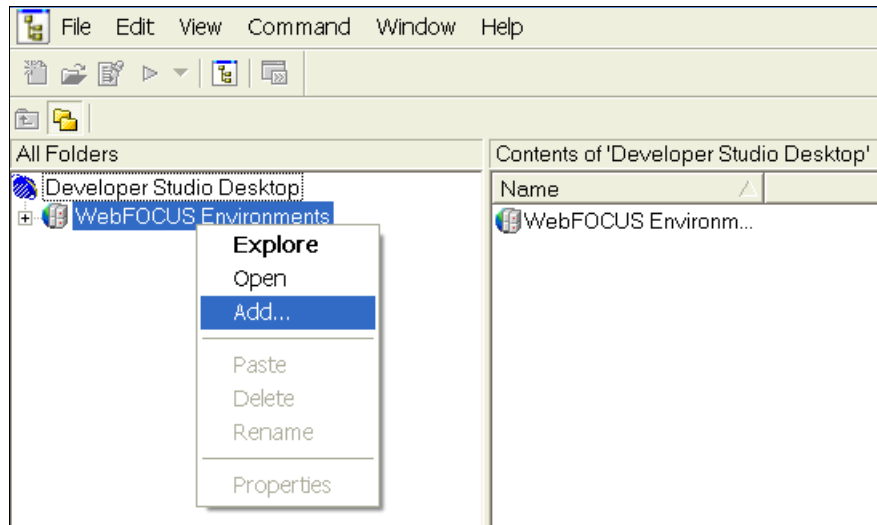


Figure 9-1 Adding System i environment

- b. In the next window (Figure 9-2), complete these steps:
- Type a description of your System i environment.
 - For Host Name/IP Address, type a name.
 - For Port, verify that 11331 is selected.

Tip: If you changed the default DB2 Web Query default port number, you must clear the **Use Default** box and specify your installation's port number.

- For HTML Alias, ensure that it is /webquery_html.
- Click **OK**.

Description:

Environment Settings (Click on object to configure)

Local Machine (JANSEN100) ↔ Web Component ↔ WebFOCUS Client ↔ Data Servers

Managed Reporting / ReportCaster ↔ Project Development

Web Component

Host Name / IP Address:

Protocol: HTML Alias: ☒ Use Default

Port:

Web Component Authentication

User ID: None

Password:

Figure 9-2 System i environment properties

3. The first time that you open the System i environment or connection that you just added, you must sign on. Sign on with your administrator or developer authorized profile. See Figure 9-3.

Tip: Select the **Remember my User Name and Password on this computer** option in this sign on and the next sign on so that you are not prompted with these sign-on windows again for this System i connection.

Click **Logon**.

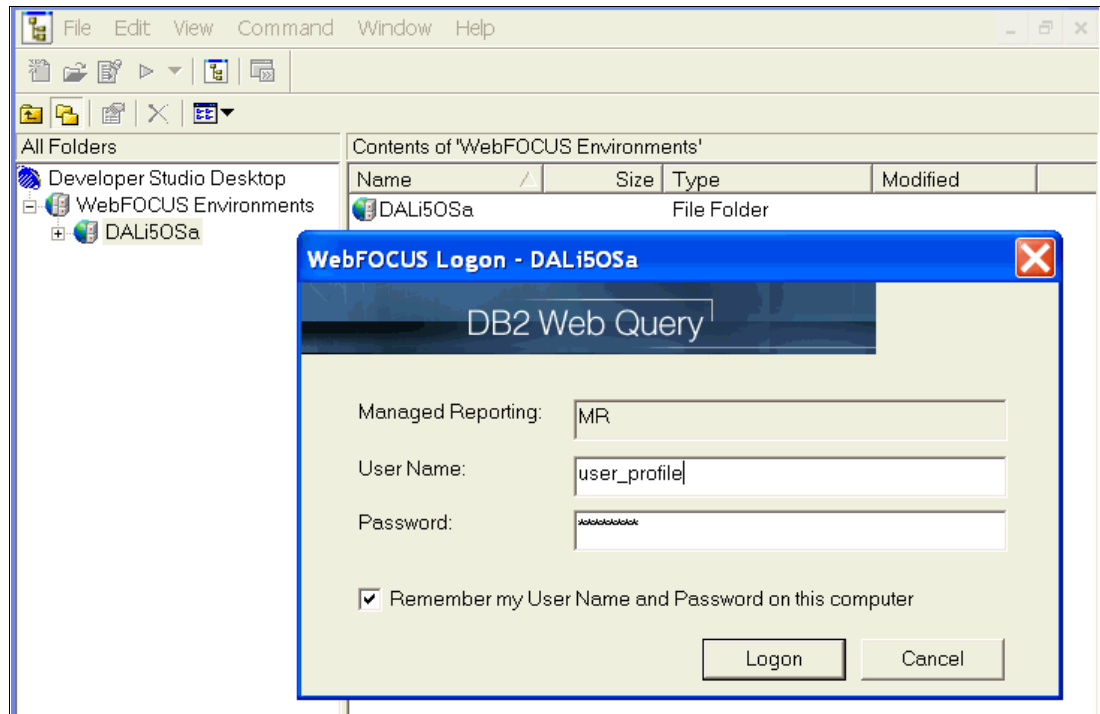


Figure 9-3 Initial opening of the System i connection

4. Expand **system name** → **Data Servers** → **EDASERVE** (Figure 9-4).
5. Log on again. Enter your user name and password and click **Logon**.

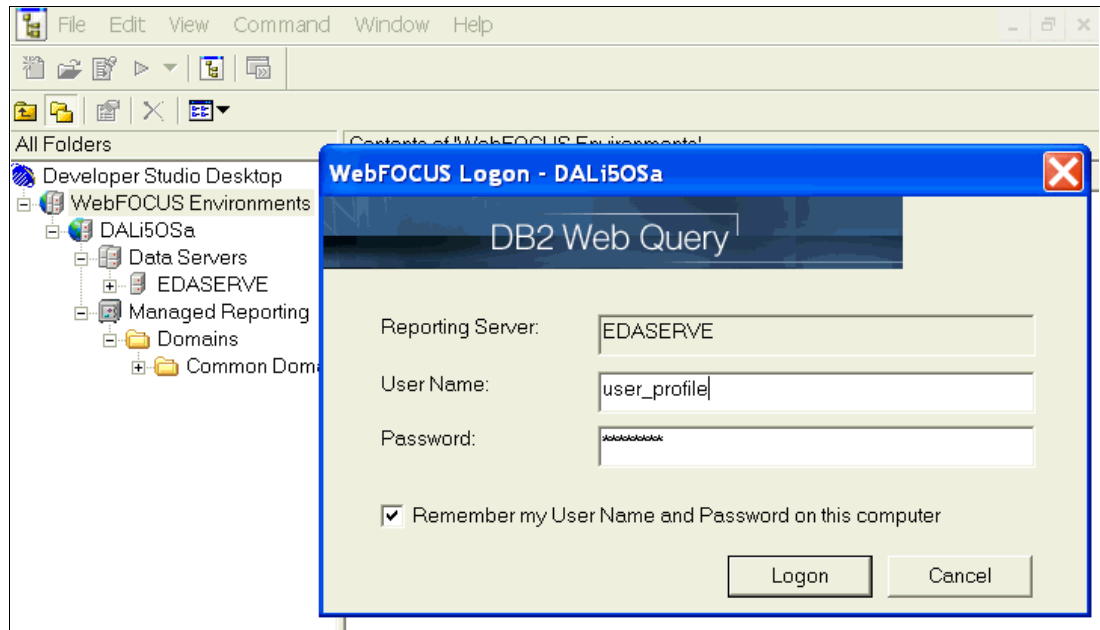


Figure 9-4 Metadata login

6. Continue down the metadata path by expanding **EDASERVE** → **Applications**. Select **baseapp**.

In the right pane of Figure 9-5, you see all your metadata or your master (.mas) and access (.acx) files. Remember that the *master file* contains the field names and formats for your table. The *access file* contains information about the actual name and location of the physical table and the primary keys for the table. Nearly all your work is done with the master file.

The source code or procedures that store your report definitions are located under the Managed Reporting → Domains portion of the tree. You expand that branch if you want to create a new report or modify an existing one.

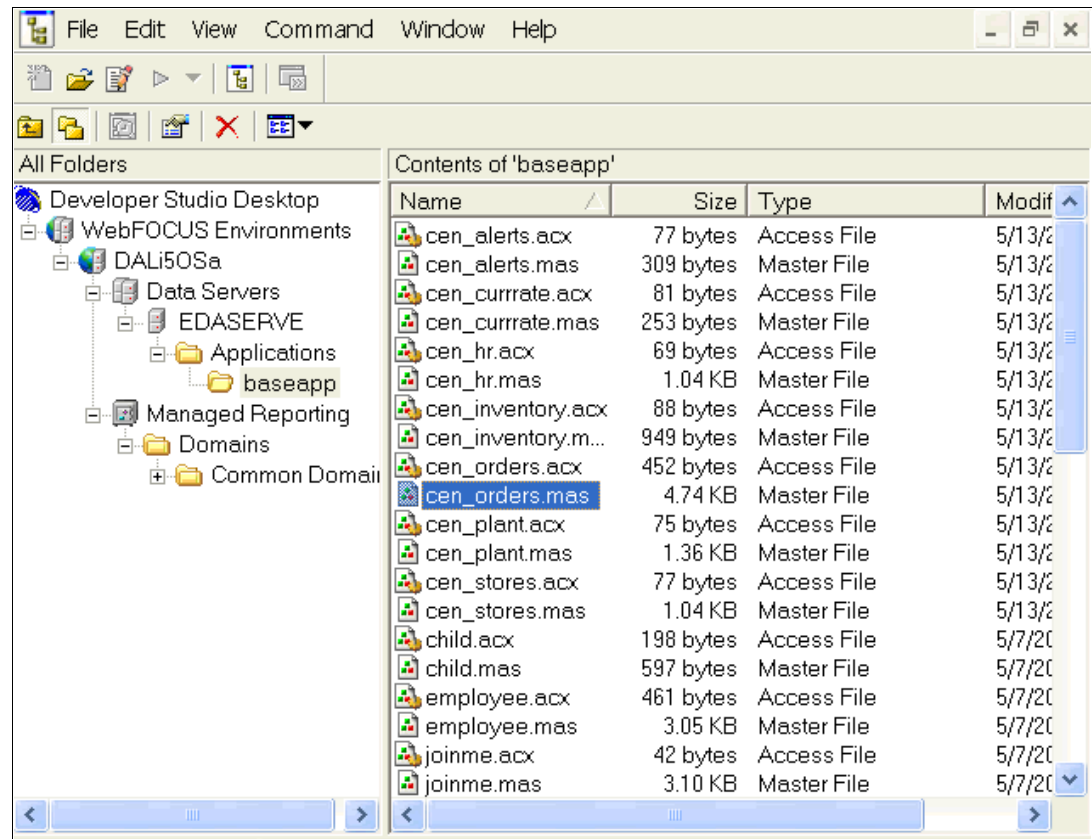


Figure 9-5 Metadata files

9.2 Setting Developer Workbench default options

Developer Workbench has a set of options that can be tailored to an individual user. To access these options, select **Window** → **Options** (Figure 9-6). We use the tailoring options in later tutorials.

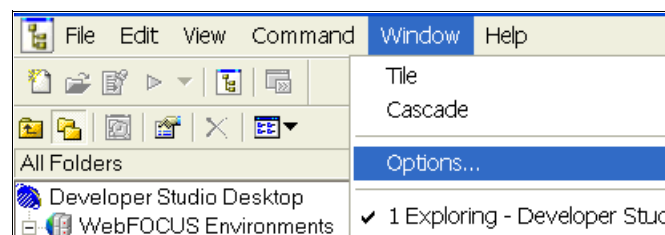


Figure 9-6 Displaying Developer Workbench options

Figure 9-7 shows the General options tab.

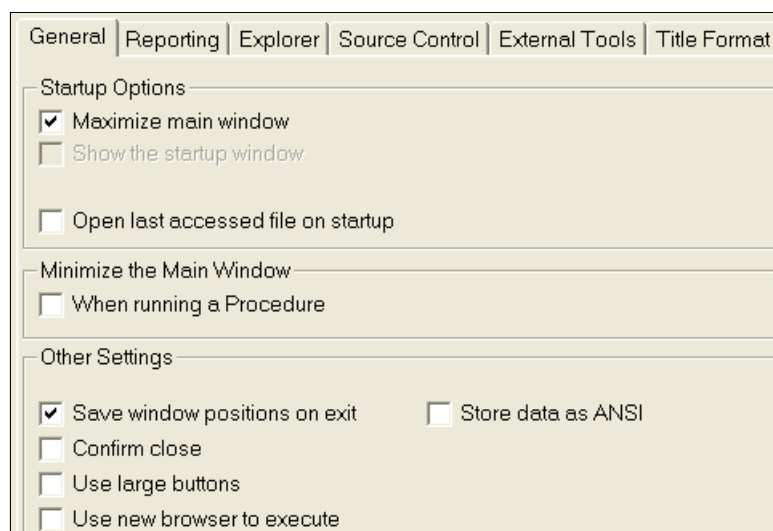


Figure 9-7 General options tab

Restriction: Report Assistant does not pick up all the reporting options set under the Reporting tab at this point in time.

Developer Workbench allows you to show your Windows desktop explorer tree in the same window as your DB2 Web Query explorer tree. On the Explorer tab, you select the Show Desktop on Explorer tree option (Figure 9-8).

Showing your Windows desktop tree within Developer Workbench has a definite benefits. The obvious benefit is the ability to stay within one working environment as a programmer. The less that you have to flip back and forth between your development mode and Windows, the better.

An additional benefit is the ease with which you can copy items from your PC hard drive to the DB2 Web Query folder structure. You might want to move style sheets, company logos, pictures, and other PC files between the different file structures.

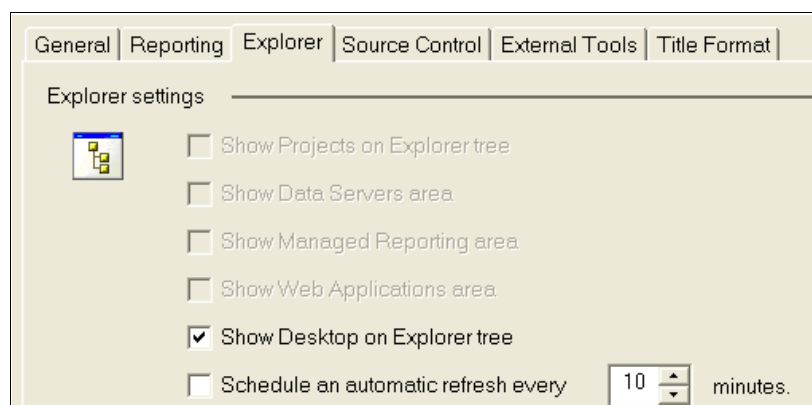


Figure 9-8 Explorer options

In Figure 9-9, we copy and paste a text file that contains SQL statements into our Other Files folder in the Common Domain. From here, the SQL can be accessed and run via the SQL Wizard in Developer Workbench or the SQL metadata creation option from the DB2 Web Query home page.

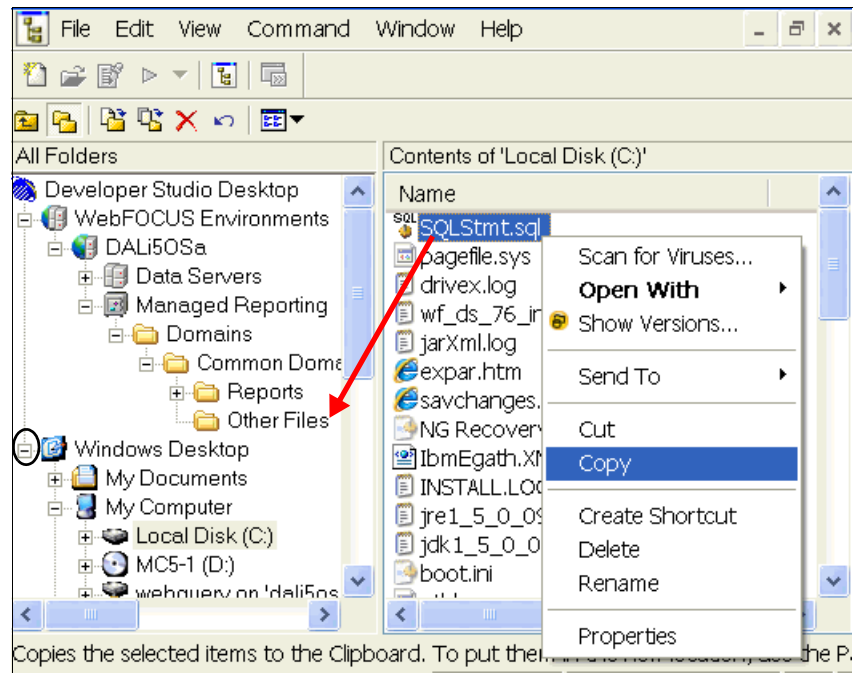


Figure 9-9 Copying files between the PC and DB2 Web Query

For more details about using Developer Workbench, refer to the online help text and the following chapters:

- ▶ Chapter 10, “Online analytical processing” on page 201
- ▶ Chapter 11, “HTML Layout Painter” on page 233
- ▶ Chapter 12, “Additional features of Developer Workbench” on page 265



Online analytical processing

Online analytical processing (OLAP) is best known as the technology that allows a user to “slice and dice” their data or drill down into their data. DB2 Web Query makes it easy to provide users with a sophisticated drill-down, slice and dice interface without the report developers having to do anything at all. You will have a better understanding of this concept by the time you finish this tutorial. After taking a standard report created with Report Assistant, the report developer selects an output option to OLAP-enable the report. This automatically provides an interface with drop-down lists and drill capable columns that allow the user to slice and dice or pivot their data and perform a variety of local analysis on the original report.

Tables must have any embedded hierarchies predefined via the Developer Workbench. After this is done, the users have an easy and intuitive way to navigate and drill down on the hierarchy within their data.

OLAP analysis typically starts with a question such as “What were my regional sales numbers across the United States?” At this point, you might look at the numbers and ask to see the details for the Eastern region which looks out of line with the other regions. If you then find that a single state is pulling down the region you might ask to see the revenue for that state split out by the different product groups you sell. After you find a product group that appears to be in trouble, you might ask to see the sales for that product group in that state summarized by month for the last two years. This type of analysis is sometimes called “Having a conversation with your data”. It is OLAP technology that enables this quick and easy interaction with the data. A user could have created a separate ad hoc query for each of the above questions but that would probably not occur in real life. If a user simply had to click a specific field of interest to go down to the next level of detail, they would be far more likely to continue with their analysis.

10.1 OLAP terminology

Table 10-1 covers basic OLAP terminology so that you understand the concepts that we use in this chapter.

Table 10-1 OLAP terminology

Term	Definition
Dimension	A set of columns structured in a hierarchy. For example, a Geographic dimension might include a City that belongs to a State, which is in a Region, which is part of a Country. In this example, Country is the top level and City is the bottom level of the dimension. A Product dimension might include Product Type at the top level and below that a Product Category, which contains multiple Models and each Model contains multiple part numbers or SKUs.
Drill down	An action that involves going from one level in a hierarchy to the next lower level. For example, if a user drills down on Country, they see the different Regions that make up the country. If they then drill down on Region, the States that compose the region are displayed. Drilling down can also involve invoking more detailed reports or graphs. For example, in an earlier tutorial, we drilled down on Product Type and passed the Product Type as a parameter to a second report. If this type of drill down is defined for a column, it takes precedence in an OLAP report over drilling down through the hierarchy.
Hierarchy	The structure or sequence of each of column within the dimension. In the example, above the Geography dimension is a hierarchy of Country, Region, State, and City.
Hyperlinks	A column that is displayed in blue in a report. The user can click the hyperlinked column to initiate an action. The standard action that is linked to a column is the ability to drill down to the next level of data. A hyperlink can also take you to another report as previously demonstrated.
Measure	Numeric fields that you want to analyze at the various levels of a hierarchy and at various intersection points. For example, you might want to analyze sales of video equipment in the United States. This is the specific intersection or slice of the data in which you are interested. Normally these fields are summed or totalled, but they do not need to be. For example, you might want to report your revenue by Country and then drill down to the revenue in the regions that make up the Country. Now the revenue is summed by Region. You might also want to see the average profit on an order across States or Regions to see if certain areas are offering too deep a discount. In this case, profit is an average measure, not a summed or totaled measure.
Pivot	To move a field from a column to a row or vice versa. For example, Year might be one of the “by” fields where the different years are listed down the side of your report. If you wanted the various years across the top of your report instead, you pivot the field or heading “Year”. You can do this in multiple ways in an OLAP report including dragging the field or using the control panel to pivot your data.

10.2 Defining OLAP metadata using Developer Workbench

Before a user can successfully OLAP enable a report, the tables used must have any existing hierarchies or dimensions defined to DB2 Web Query.

Important: Prior to reading this section, you must read Chapter 9, “Developer Workbench” on page 193.

Before a user can OLAP enable a report, a developer must define any hierarchies that are present in the data. This hierarchy is automatically used by DB2 Web Query whenever any future user asks to OLAP enable a report that references tables that contain these hierarchies. *Hierarchies*, also known as *dimensions*, can involve elements from multiple tables, although the norm is to have a single hierarchy composed of columns from a single table.

1. Start Developer Workbench.
2. Expand **DB2 Web Query Environments** → **system name** → **Data Servers** → **EDASERVE** → **Applications** → **Baseapp**.
3. Remember that the term *synonym* is synonymous with *metadata*. To define a hierarchy, edit the metadata for a table using the Synonym Editor.
 - a. Right-click the **cen_orders.mas** table and select **Edit in Synonym Editor** (Figure 10-1).

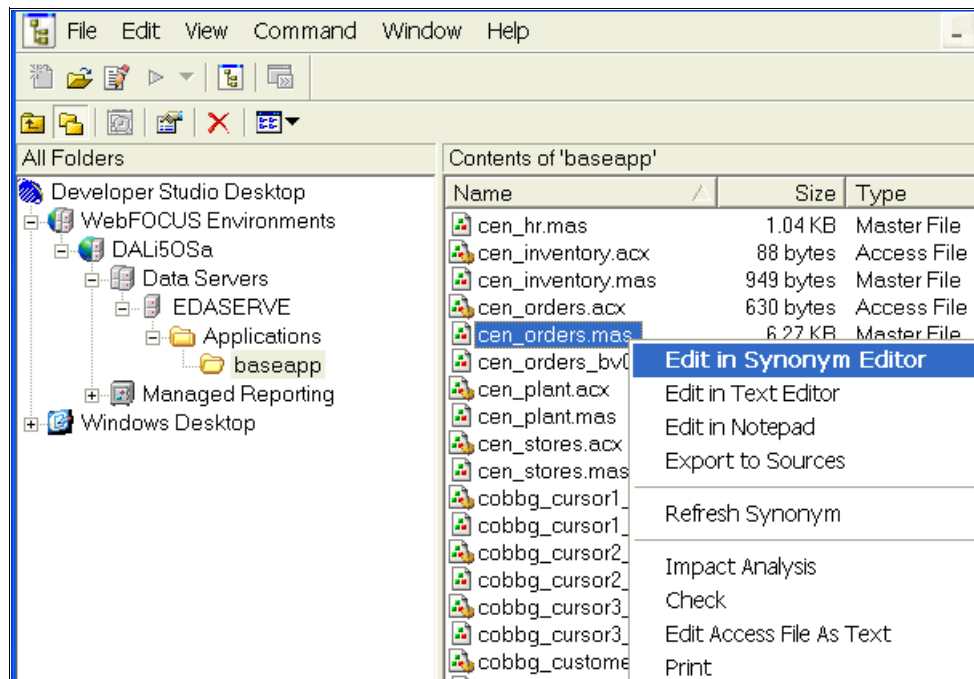


Figure 10-1 Developer Workbench metadata

- At the bottom of the window that opens (Figure 10-2), click the **Modeling View** button. This is the easiest way to define hierarchies to DB2 Web Query.

Note: In the left pane of this window are four tables pane—Orders, Inventory, Plant, and Stores—even though you only opened the master file or metadata for the Orders table. This tells you that the Orders table was defined in the System i environment with foreign keys pointing to the Inventory, Plant, and Stores tables. When you do this, DB2 Web Query automatically pulls in the metadata for all four tables into one “clustered” table.

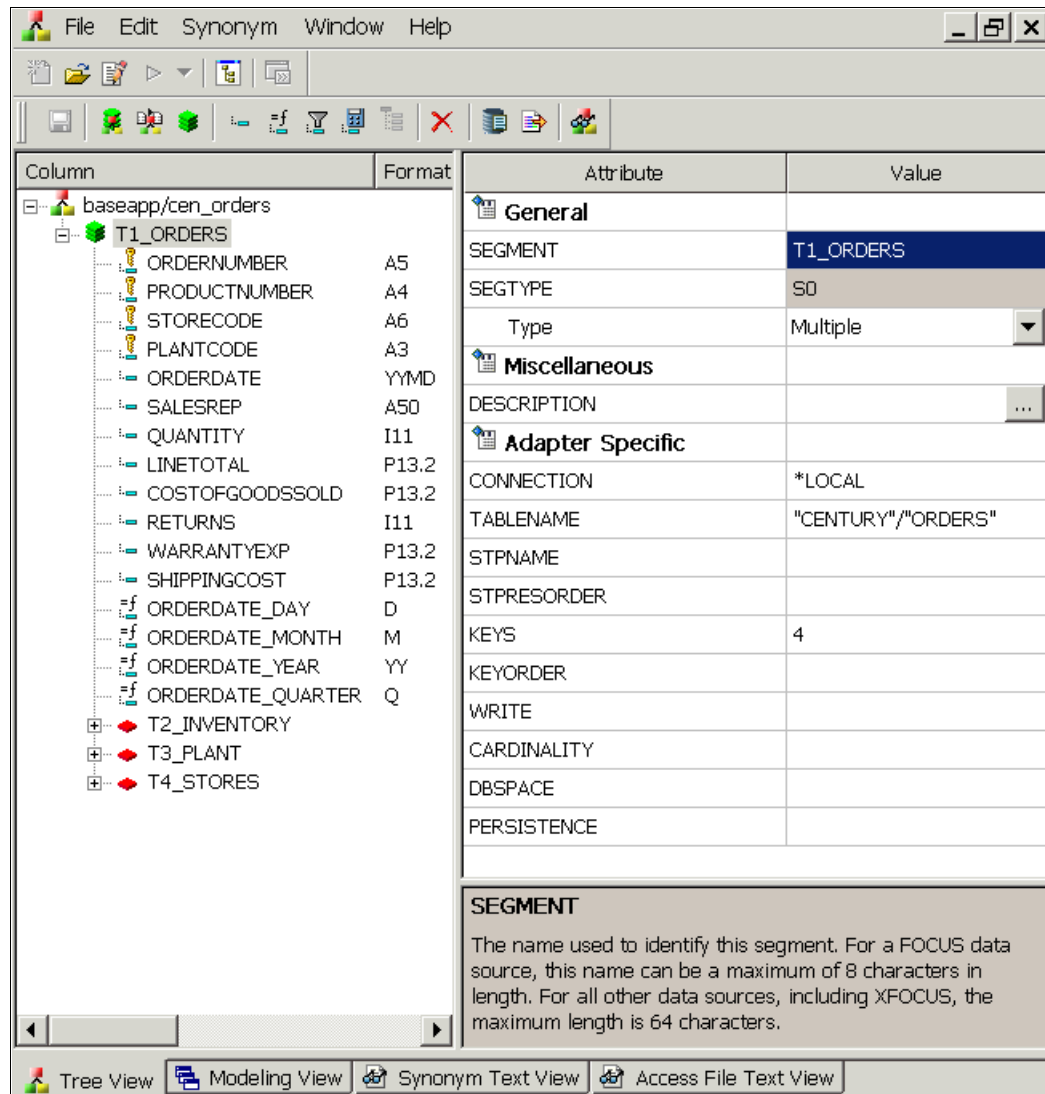


Figure 10-2 Developer Workbench Synonym Editor

5. If Dimension Builder is not visible, click **Synonym** → **Dimension Builder**. Figure 10-3 shows the Dimension Builder pane. To the right of the Dimension Builder view (Figure 10-3), you see the tables and their relationship to the base table Orders.

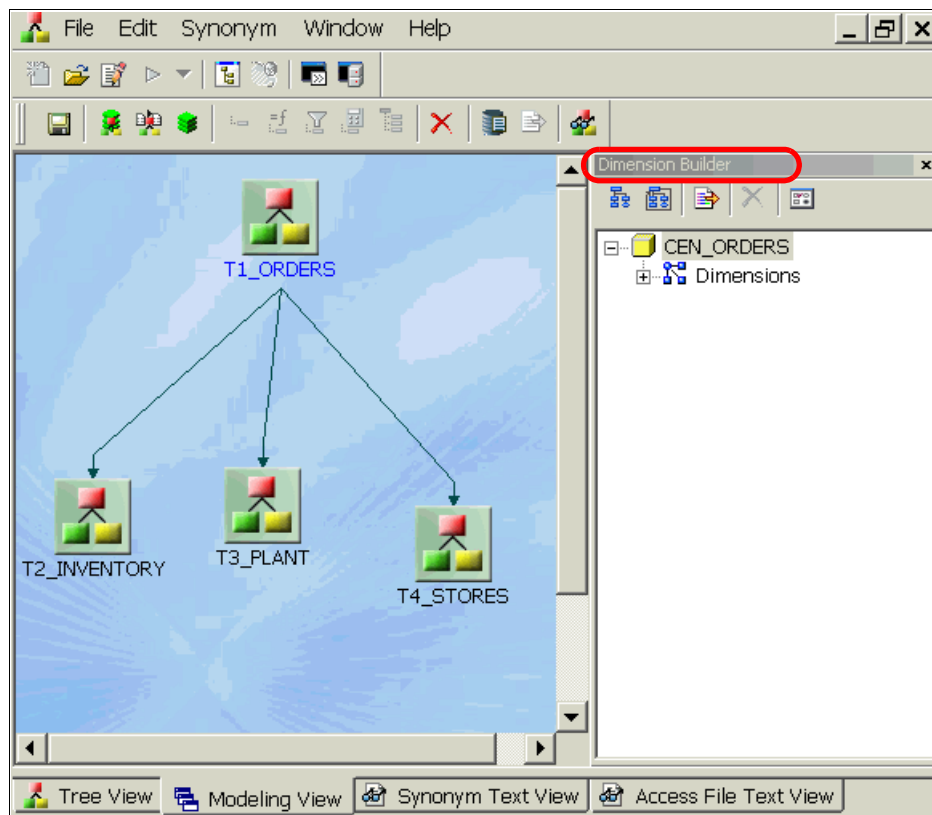


Figure 10-3 Dimension Builder

- You can either view the tables as icons or view a column list for each table. Double-click the **Inventory** table icon to switch to column view (Figure 10-4). You can double-click the icon again to revert back to icon mode.

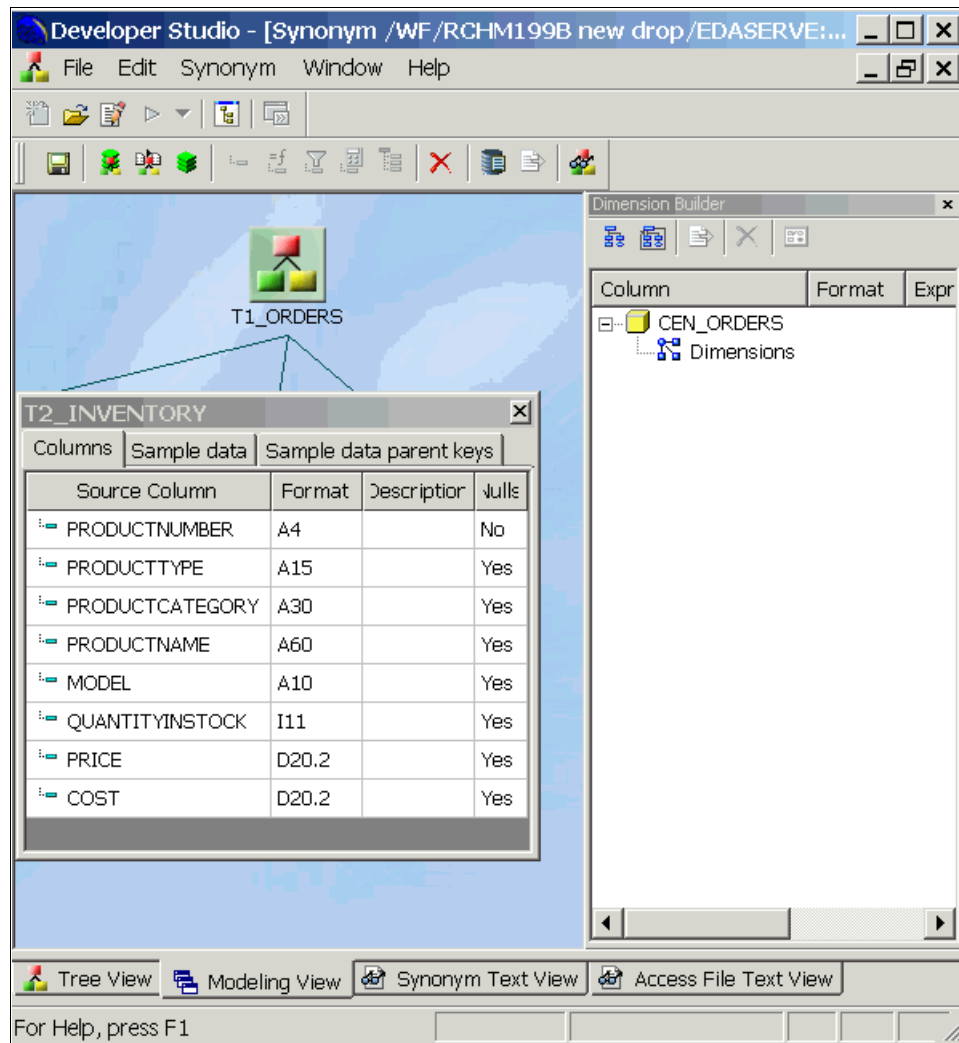


Figure 10-4 Icon or detailed table view

- The columns in the Inventory table are structured so that, within product categories, you have product types; within product types, you have models; and within models, you have product names. We must describe this hierarchy to DB2 Web Query.

Right-click **Dimensions** and select **Add Hierarchy** → **Levels** (Figure 10-5 on page 207).

Tip: A quicker way to achieve the same result is to click the **hierarchy** icon (circled in Figure 10-5) at the top of the Dimension Builder pane.

There are two types of hierarchies one is based on levels and the other is based on parent child relationships. We use level hierarchies. The level hierarchy is the example that we have used so far; that is Country, Regions, State, and City are a level hierarchy.

For more information about the different types of hierarchies, see the Developer Workbench help text.

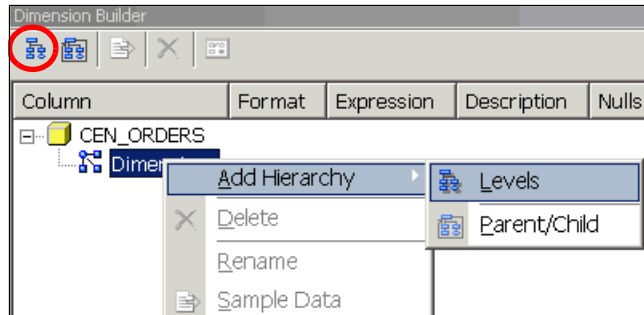


Figure 10-5 Adding a hierarchy

8. Right-click your new **dimensions** icon and select **Rename**. Change the name to Product Hierarchy.
9. Starting at the top of the hierarchy (Product Type), drag the column name to Product Hierarchy. Repeat this action for Product Category and Product Name (Figure 10-6).

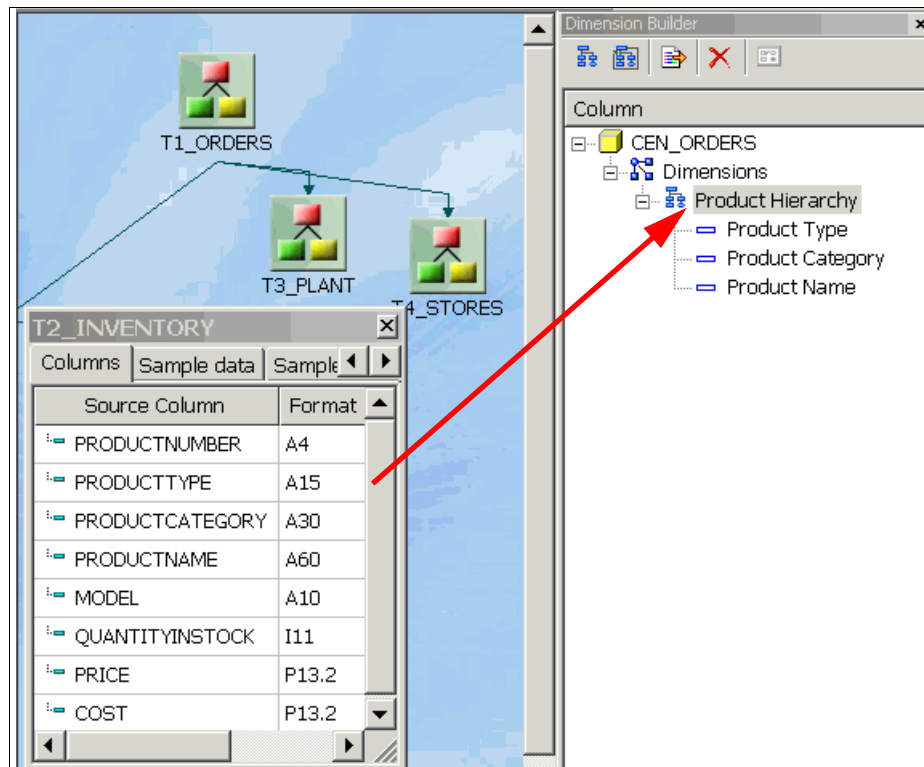


Figure 10-6 Product Hierarchy

10. Define the Date Dimension shown in Figure 10-7:

- Right-click **Dimensions** and select **Add Hierarchy** → **Levels**
- Right-click your new **dimensions** icon and select **Rename**. Change the name to Date Dimension.
- Drag the column name to the Date Dimension for ORDERDATE_YEAR, ORDERDATE_QUARTER, ORDERDATE_MONTH, and ORDERDATE_DAY.

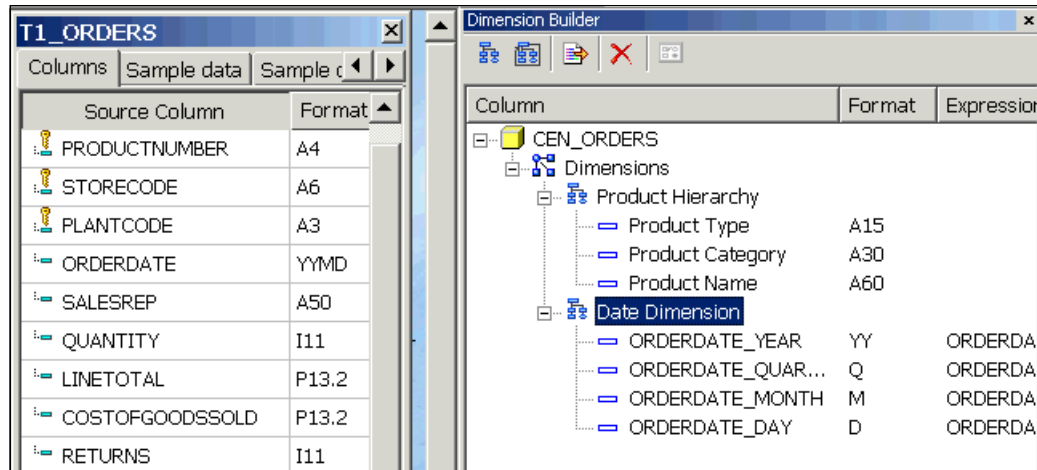


Figure 10-7 Date Dimension

11. Define the Plant Hierarchy as shown in Figure 10-8 on page 209:

- Right-click **Dimensions** and select **Add Hierarchy** → **Levels**
- Right-click your new **dimensions** icon and select **Rename**. Change the name to Plant Hierarchy.
- Drag the column name to Plant Hierarchy for Plant Country, Plant Region, Plant State, and Plant City.

12. Define the Store Hierarchy as shown in Figure 10-8 on page 209.

- Right-click **Dimensions** and select **Add Hierarchy** → **Levels**
- Right-click your new **dimensions** icon and select **Rename**. Change the name to Store Hierarchy.
- Drag the column name to Store Hierarchy for COUNTRY, REGION, STATE, CITY, and Store Name.

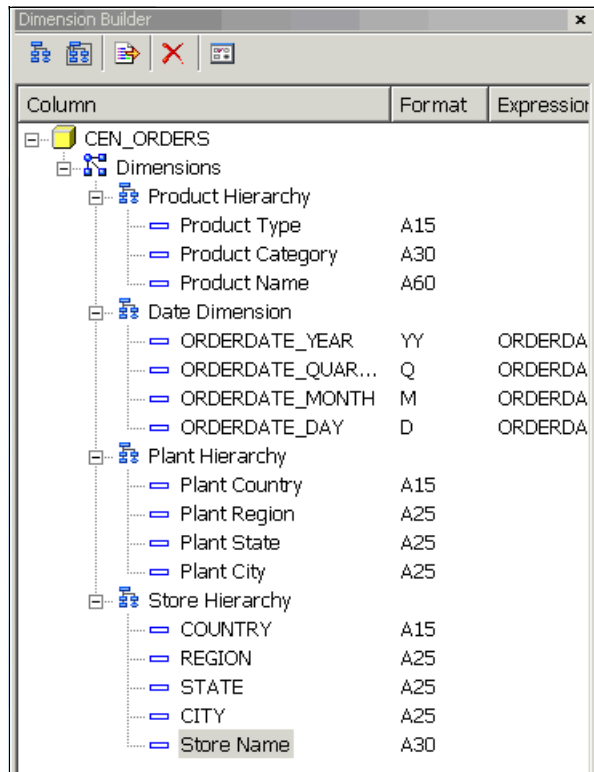


Figure 10-8 Developer Workbench Multiple hierarchies for OLAP

13. If you do not see the additional date fields such as ORDERDATE_YEAR, decompose your date as explained in 12.1.1, “Date Decomposition” on page 269.

14. Save your definition.

You have now completed all the definitions that are required for OLAP-enabling your reports. Typically this is done once by the IT department. After this, any user with authority to create reports can choose to OLAP-enable their report.

You can see how easy it is to define a hierarchy in your data and create the necessary metadata to OLAP-enable your reports for detailed analysis.

Note: OLAP is an optional feature with DB2 Web Query. As you have just seen, Developer Workbench, also an optional feature, is a prerequisite for creating OLAP-enabled reports. Developer Workbench is the only interface for defining the hierarchies that are required by OLAP.

10.3 Enabling an OLAP report (OL1_Revenue)

To enable an OLAP report:

1. Navigate to your **Tutorials** folder, right-click the **RA1_Revenue** report, and select **Open**.
2. Click the **Report options** tab.
3. On the Report options tab, complete these steps:
 - a. Since we know this is a small report, for Page numbering, select **Off** (Figure 10-9).

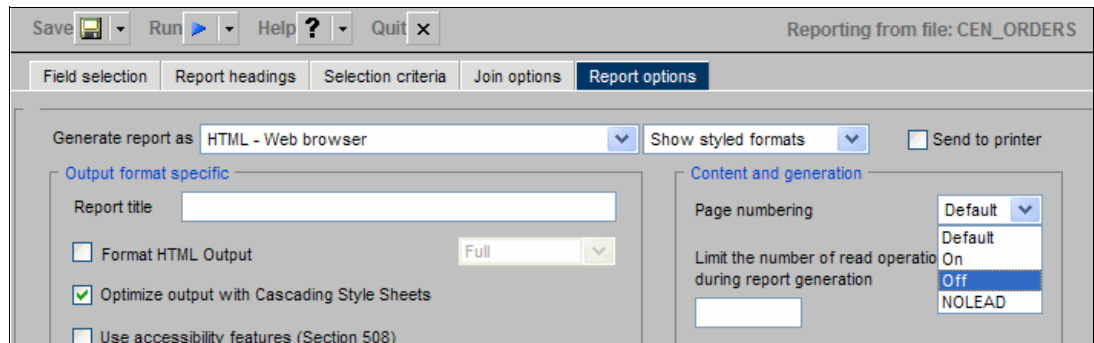


Figure 10-9 Turning off default page numbering

- b. Under the Enable OLAP, select **Show Tabbed**.

Enable OLAP options: The Enable OLAP options control how users can interact with an OLAP report and access OLAP tools:

- ▶ **Off:** Provides an OLAP button to open the OLAP Control Panel. There is no top Selection pane.
- ▶ **Columns only:** Allows you to sort, hide, drag and drop columns. There is no top Selection pane and there is no access to the OLAP Control Panel.
- ▶ **Columns with panel:** Looks the same as Columns only, but when you click the square blue button over a dimension, you open the OLAP Control Panel.
- ▶ **Show filters on top/bottom:** Adds the OLAP Selections pane to the top (or bottom) of the report. This option is acceptable with a few dimensions and a few levels in each hierarchy. It tends to look messy when you have too many levels within different dimensions. Try it to see the difference between showing dimensions and hierarchies as filters or as tabs.
- ▶ **Show Panel in Report:** Is the same as the Columns with panel option but includes a small OLAP symbol in the lower left to more obviously allow the user to open the OLAP Control Panel.
- ▶ **Show Tabbed:** For OLAP reports that have multiple dimensions, this option groups the dimension elements under a tab labeled with the dimension name. This is the option we use.

- c. Under the Automatic drill down, select **Dimensions and Measures** (Figure 10-10). This gives you the maximum drill-down capabilities. When creating your own reports, you can choose to drill down on the dimensions only and not the measures.

Automatic drill down options:

- **None:** The user cannot drill down to a lower level in the hierarchy within the report itself.
- **Dimensions:** The user can drill down on a dimension to proceed to the next level down in the hierarchy.
- **Dimensions and Measures:** In addition to drilling down on the dimensions, the user can drill down on the measures within the report. This allows the users to drill down to the lowest level of all displayed hierarchies.

The screenshot shows the 'Report options' tab of a report configuration interface. The 'Generate report as' dropdown is set to 'HTML - Web browser (with grid)'. The 'Show styled formats' dropdown is set to 'Default'. Under 'Output format specific', 'Format HTML Output' is unchecked, 'Optimize output with Cascading Style Sheets' is checked, and 'Use accessibility features (Section 508)' is unchecked. The 'Summary text for report' field is empty. Under 'Interactive options', 'Enable OLAP' is set to 'Show Tabbed', 'Automatic drill down' is set to 'Dimensions and Measures', 'On-demand paging' is unchecked, and 'Show pop-up field description' is unchecked. Under 'Global report styling', 'Font...' is a button, 'Active Report Styling' is a button, 'Apply an existing WebFOCUS Style Sheet' is set to 'ibm_blue1', and 'Apply an existing Cascading Style Sheet' is set to 'NONE'. On the right, 'Page numbering' is set to 'Default', and 'Limit the number of read operations performed during report generation' is set to '0'. 'Stop retrieving records after the specified number have been included in the report' is set to '0'. Under 'Database Output', 'Destination/Library' is a text field, 'Name' is set to 'HOLD', and 'Overwrite existing file' is unchecked.

Figure 10-10 OLAP enabling a report

4. Save your report as OL1_Revenue.

You have now OLAP enabled your report. To summarize, we took a basic report and selected an output option to enable OLAP. This simple selection automatically gives you an enormous amount of analysis capabilities as you will see in the next section.

10.4 Working with an OLAP-enabled report

To begin working with the OLAP-enabled report as explained in this tutorial, run your **OL1_Revenue** report. Figure 10-11 shows this report.

Note: If you run your report from the DB2 Web Query home page, there is a maximize icon to the far right. This sends your report to a new browser window.

Notice the addition of a completely new panel at the top of the report. This is called the *Selection pane*. It includes tabs for every dimension described for any of the tables that are part of the Orders cluster. You also have the capability to work with the measures and to create graphs directly from the report.

There are three ways to work with your OLAP data:

- ▶ From the report itself
- ▶ From the Selections pane
- ▶ From the Control Panel

In the report shown in Figure 10-11, you cannot see the control panel. In this example, the control panel starts as hidden or closed. You can open it by clicking the OLAP button or by clicking the blue square to the left of the Product Type heading.

Product Type	Order Count	Revenue
Audio	12.846	\$382,683,321.00
Camcorders	5.139	\$444,531,041.00
Cameras	5.307	\$184,103,667.00
Office	2.958	\$30,245,685.00
Video	6.033	\$520,360,205.00
TOTAL	32,283	\$1,561,923,919.00

Figure 10-11 OL1_Revenue report

10.4.1 Slicing, dicing, and drilling down for more details

Looking at the report in Figure 10-11, notice that the heading for every column either has a blue square or a set of arrows. The blue square allows you to drag and drop the column and to rearrange or pivot it. It also allows you to open up the OLAP Control Panel. The arrows allow you to sort the measures or numeric fields.

Tip: When working with an OLAP-enabled report, there are multiple ways to accomplish the same end result.

To use the arrows and buttons to drill down for more details, follow these steps:

1. Because Revenue was previously unsorted, neither half of the arrow is dark and the prompt reads “sort Revenue”. Click the **sort arrow** to the left of Revenue (Figure 10-12).

 Product Type	 Order Count	 Revenue
Audio	12,846	\$382,683,321.00
Camcorders	5,139	\$444,531,041.00
Cameras	5,307	\$184,103,667.00
Office	2,958	\$30,245,685.00
Video	6,033	\$520,360,205.00
TOTAL	32,283	\$1,561,923,919.00

Figure 10-12 Sorting a measure

2. Now that you have sorted by Revenue, hover over the column again. Notice that the bottom half of the arrow is dark. Click this arrow, and you sort the column in descending sequence (Figure 10-13).

 Product Type	 Order Count	 Revenue
Office	2,958	\$184,103,667.00
Cameras	5,307	\$184,103,667.00
Audio	12,846	\$382,683,321.00
Camcorders	5,139	\$444,531,041.00
Video	6,033	\$520,360,205.00
TOTAL	32,283	\$1,561,923,919.00

Figure 10-13 Sorting from highest to lowest

Here is another way to sort Revenue. Right-click the column heading **Revenue** and select **Visualize** (Figure 10-14).

 Product Type	 Order Count	 Revenue
Video	6,033	
Camcorders	5,139	
Audio	12,846	
Cameras	5,307	
Office	2,958	
TOTAL	32,283	\$1,561,923,919.00

Figure 10-14 Revenue context menu

As we saw in Active Reports, selecting Visualize displays bars to help you quickly note trends or to easily compare two columns. See Figure 10-15.

Note: Positive numbers are visualized with blue bars, while negative numbers are visualized with red bars.

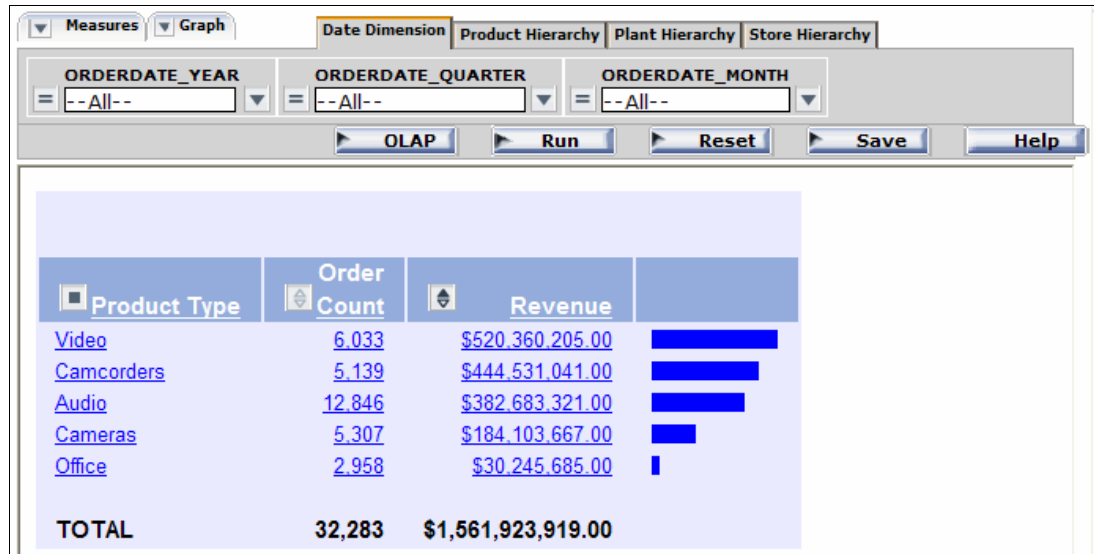


Figure 10-15 Revenue visualization

- Click the **Store Hierarchy** tab.

The Store Hierarchy tab has a drop-down tab for each element in the hierarchy, proceeding from the top of the hierarchy to the bottom.

By default, each of the drop-down lists contains all the unique values for that column. For example, initially every region in the world is available under the Region tab.

- To start to filter and drill down on your data, click the **Country** drop-down list and select **United States** (Figure 10-16).

Tip: You can select multiple values using the Ctrl or Shift keys.

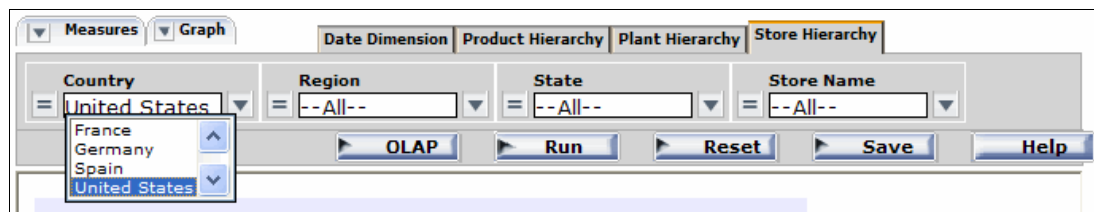


Figure 10-16 Selecting a hierarchy

- Click the **Region** drop-down list. Notice that the only regions displayed are for the United States. Select the **North** region (Figure 10-17).

Figure 10-17 Continuing to filter within hierarchy

- The filters in the Selection pane do not take effect until you click Run. Review your current totals and click **Run** (Figure 10-18).

Your report now shows the revenue and number of orders for the North region in the United States.

Figure 10-18 Run your filtered report

- Notice that immediately to the right of COUNTRY and REGION is an equal sign (=). Click the = button. Each time you fill in a value, you filter your data by creating a WHERE clause. Instead of selecting COUNTRY equal to United States, we can select all countries except the United States by changing the button to not equal. Click the button multiple times to cycle through your choices.
- Click the **Reset** button to reset the report back to its initial state. You can also use the back arrow on our browser to go back one page at a time.
- Click the **Product Hierarchy** tab (Figure 10-19).

Product Type	Order Count	Revenue
Audio	12,846	\$382,683,321.00
Camcorders	5,139	\$444,531,041.00
Cameras	5,307	\$184,103,667.00
Office	2,958	\$30,245,685.00
Video	6,033	\$520,360,205.00
TOTAL	32,283	\$1,561,923,919.00

Figure 10-19 Drilling down on Product Type

10. Drill down within the report, not by using the hierarchy tabs. Click **Audio**.

Notice that not only did you drill down from Product Type to the next level of Product Categories within Audio, DB2 Web Query also filled in the Selection pane tabs for you (Figure 10-20).

Product Category	Order Count	Revenue
Amplifiers/PreAmps/Tuners	1,552	\$42,374,428.00
Audio Systems	1,384	\$122,345,680.00
CD Players and Recorders	1,306	\$53,847,459.00
MP3	3,320	\$43,491,588.00
Receivers	1,337	\$35,907,113.00
Speakers	3,947	\$84,717,053.00
TOTAL	12,846	\$382,683,321.00

Figure 10-20 Drill down on Product Category

11. Drill down on **Amplifiers/PreAmps/Tuner** to see the products that compose the Amplifier category.

Again notice that the drop-down lists have been filled in for you. You can go back and forth this way between the body of the report and the tabs at the top. See Figure 10-21.

Product Name	Order Count	Revenue
AM / FM Stereo Tuner	90	\$950,822.00
Modular Components Series Preamp 5.1	222	\$6,376,419.00
Power Amplifier	315	\$6,116,685.00
PreAmp/Tuner Two	697	\$24,364,174.00
PA4000 Stereo & Surround Power Amplifier	228	\$4,566,328.00
TOTAL	1,552	\$42,374,428.00

Figure 10-21 Results of Product Category drill down

12. Continuing to analyze our data, drill down to see only the Amplifier sales that occurred in January 2007. Click the **Date Dimension** tab.

13. For Year, select **2007**. You can then select the quarter or go straight to month and select **01** (Figure 10-22).

Figure 10-22 Selecting values in non-consecutive levels

14. Click **Run**. Figure 10-23 shows the results of running this report.

Product Name	Order Count	Revenue
AM / FM Stereo Tuner	1	\$199.00
Modular Components Series Preamp 5.1	3	\$9,975.00
Power Amplifier	6	\$84,909.00
PreAmp/Tuner Two	13	\$330,837.00
PA4000 Stereo & Surround Power Amplifier	5	\$162,955.00

Figure 10-23 Slice of your data (January 2007)

15. Click **Reset** to start back at the beginning with an unfiltered report.

16. Try drilling down on a measure or total field. Drill down on the **Revenue** column for **Office** products (Figure 10-24).

Product Type	Order Count	Revenue
Audio	12,846	\$382,683,321.00
Camcorders	5,139	\$444,531,041.00
Cameras	5,307	\$184,103,667.00
Office	2,958	\$30,245,685.00
Video	6,033	\$520,360,205.00

Figure 10-24 Drilling down on a measure

After drilling on the revenue for Office products, the Office product type is expanded to its subcategories and products. The Revenue and Order count are now grouped at a much more detailed level (Figure 10-25).

Product Category	Product Name	Order Count	Revenue
Handheld and PDA	Handheld PDA 32MB Memory	287	\$7,398,456.00
	Handheld PDA 64MB Memory	95	\$3,572,646.00
	Pocket PC Handheld Organizer 128MB Memory	94	\$3,308,508.00
	Pocket PC Handheld Organizer 32MB Memory	144	\$2,733,464.00
	Pocket PC Handheld Organizer 64MB Memory	62	\$1,520,116.00
Organizers	Easyfile Electronic Organizer 10MB Memory	206	\$934,381.00
	Easyfile Electronic Organizer 8MB Memory	621	\$2,105,259.00
	Easyvoice Voice Recorder 6 Hours	265	\$1,381,072.00
	Easyvoice Voice Recorder 8 Hours	989	\$5,999,343.00
	Easyvoice VoiceEmail - Digital Recorder	195	\$1,292,440.00
TOTAL		2,958	\$30,245,685.00

Figure 10-25 Results of drilling down on a measure

17. Reset your report now so that all drill downs are removed.

18. The next request from our users is to generate a report that shows revenue by product type by year.

Drag **ORDERDATE_YEAR** from the Selections pane into the area of the report where you want it, that is across the top. The plus symbol (+) shows that you are in a valid area to drop the field (Figure 10-26).

Product Type	Order Count	Revenue
Audio	12,846	\$382,683,321.00
Camcorders	5,139	\$444,531,041.00
Cameras	5,307	\$184,103,667.00
Office	2,958	\$30,245,685.00
Video	6,033	\$520,360,205.00
TOTAL	32,283	\$1,561,923,919.00

Figure 10-26 Dragging and dropping dimensions

19. Right-click **Order Count** and select **Hide** (Figure 10-27).

ORDERDATE_YEAR		2006		2007	
Product Type	Order Count	Revenue	Order Count	Revenue	
Audio		920.684.00	6.348	\$191.762.637.00	
Camcorders		517.006.00	2.573	\$228.014.035.00	
Cameras		761.413.00	2.593	\$93.342.254.00	
Office		658.888.00	1.467	\$15.586.797.00	
Video		444.717.00	3.017	\$266.915.488.00	

Figure 10-27 Hiding a field

20. We now have a pivot table. If you decide that you prefer to see the years down the left column, drag the column name **ORDERDATE_YEAR** to the left of Product Type (Figure 10-28). Again, the + sign directs you to where you can validly drag the field.

ORDERDATE_YEAR		2006		2007	
Product Type	Revenue	Revenue			
Audio	\$190.920.684.00	\$191.762.637.00			
Camcorders	\$216.517.006.00	\$228.014.035.00			
Cameras	\$90.761.413.00	\$93.342.254.00			
Office	\$14.658.888.00	\$15.586.797.00			
Video	\$253.444.717.00	\$266.915.488.00			
TOTAL	\$766,302,708.00	\$795,621,211.00			

Figure 10-28 Pivot report

21. Click the **Save** button and select **Save Report** to save the new pivot report (Figure 10-29). You can also see what happens when you save to the Excel and PDF output formats.

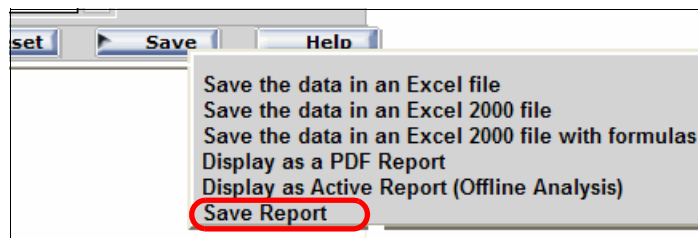


Figure 10-29 Saving an OLAP report

22. In the window that opens (Figure 10-30), specify the folder for your report. If you want to save this report without any OLAP capabilities, select the Save without OLAP option. Since you used the OLAP functionality to design a new report for a user, we do not select this option. Click **OK**.

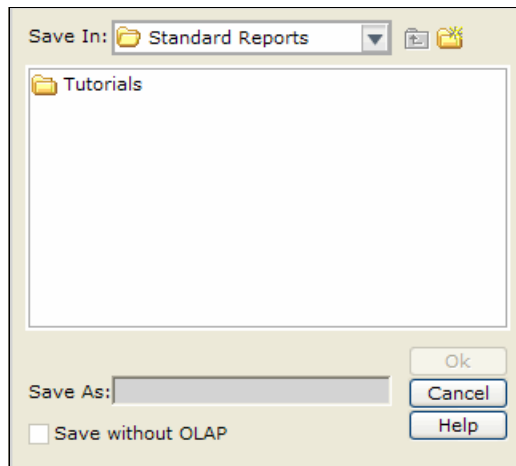


Figure 10-30 Saving a report to your folder

23. Our users have changed their minds and decided that they do not want a pivot table. Instead they want to see the year and product type to the left of Revenue. Drag **ORDERDATE_YEAR** from the Sort across portion of the report to the Sort by section (Figure 10-31).

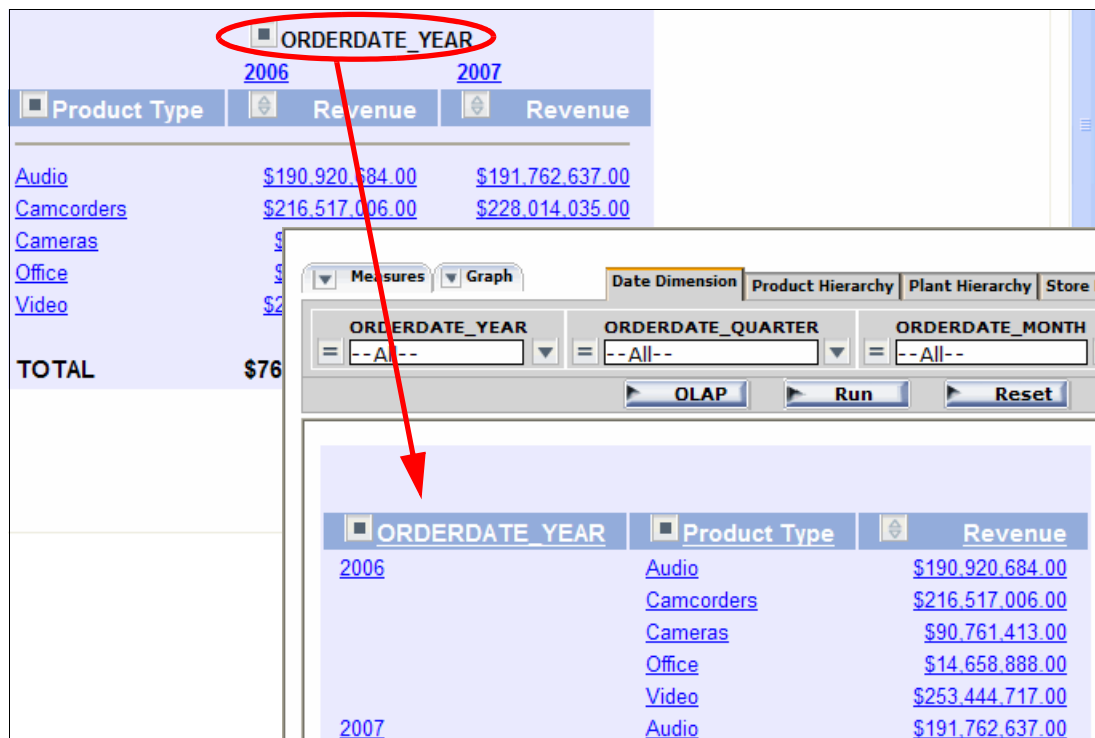


Figure 10-31 Pivoting a column

In addition to dragging from the Selection pane area, there are other methods to display additional columns in the report as explained in the following steps.

24. Right-click **ORDERDATE_YEAR** and select **Unhide** (Figure 10-32). You see a list of columns that were either part of the initial report or are in one of the hierarchies. These are the columns that are available for display.

ORDERDATE_YEAR	Product Type	Revenue
2007	Audio	\$190,920,684.00
	ORDERDATE_YEAR	\$216,517,006.00
	ORDERDATE_QUA...	\$90,761,413.00
	ORDERDATE_MON...	\$14,658,888.00
	Product Type	\$253,444,717.00
	Product Category	\$191,762,637.00
	Product Name	\$228,014,035.00
	Plant Country	\$93,342,254.00
	Plant Region	\$15,586,797.00
	Plant State	\$266,915,488.00
	Plant City	
	Country	
	Region	
	State	
	Store Name	
TOTAL		\$1,561,923,919.00

Figure 10-32 Unhiding columns

25. Try the same process for the Revenue field. Right-click **Revenue** and select **Unhide**. Notice that you see that only numeric fields are available.

26. Click the down arrow for the **Measures** tab (Figure 10-33):

- You see a list of all the measures that are available to you in this report. Notice that Order Count has no check mark. This is because we hid it. To display **Order Count**, select it again.
- Select **Revenue**. Notice that you can cycle through the different options for displaying, visualizing, and hiding the measure. We visualize Revenue, display Order Count, and run the report again.

ORDERDATE_YEAR	Product Type	Order Count	Revenue
2006	Audio	6,498	\$190,920,684.00
	Camcorders	2,566	\$216,517,006.00
	Cameras	2,714	\$90,761,413.00
	Office	1,491	\$14,658,888.00

Figure 10-33 Measures tab

27. Click the down arrow for the **Graph** tab (Figure 10-34):

- Select **Revenue**.
- Click the icon to the right of Revenue to cycle through the graph types until you come to pie charts.
- Click **Run**.
- Make the pie chart window larger by dragging down the horizontal break line between the graph and the report.

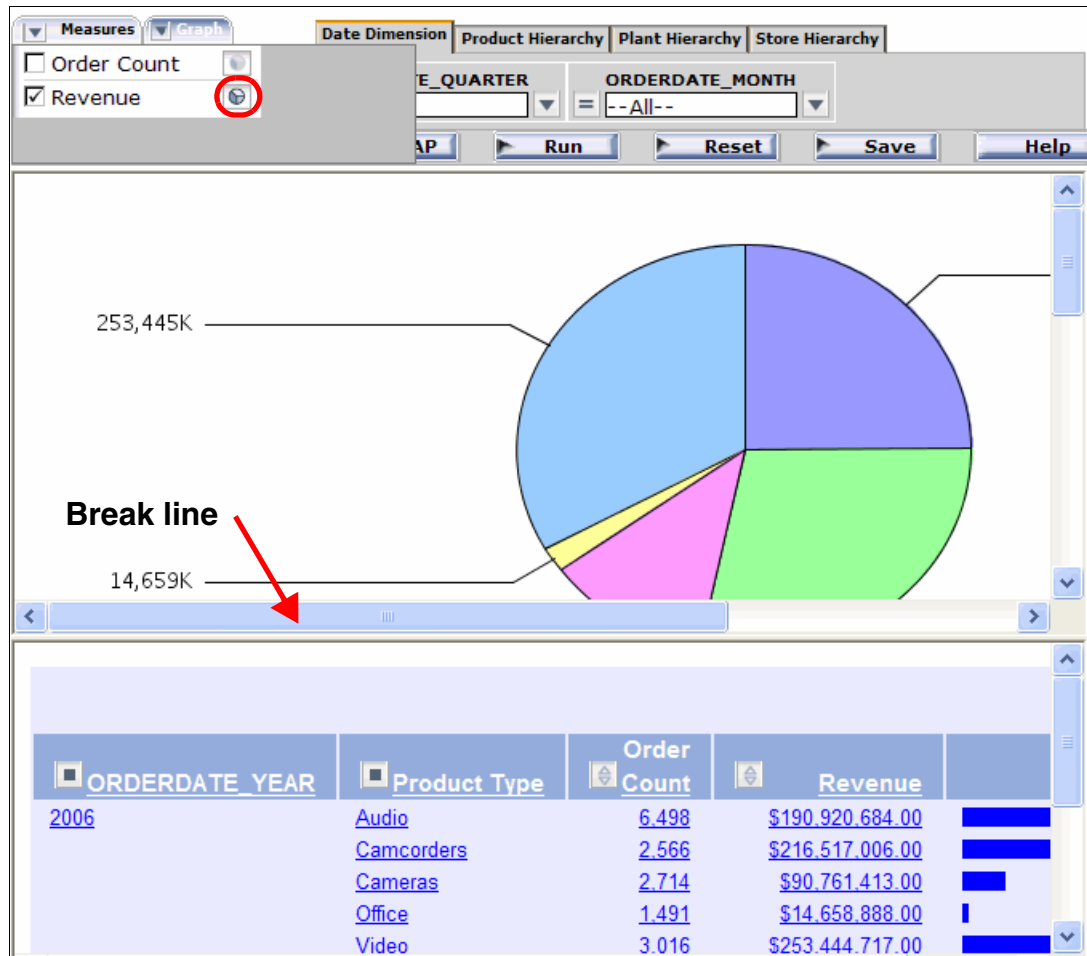


Figure 10-34 Graphing an OLAP report

Notice that there is one pie chart for each year that was your major sort by axis. See Figure 10-35.

Tip: To control the scope of the graph, use the drop-down lists in the Selection pane. For example, if you only want a graph for 2007 revenue, you can select the year 2007 from the ORDERDATE_YEAR drop-down list.

If the measures in the original report were drill down capable, then the graph can be drilled into.

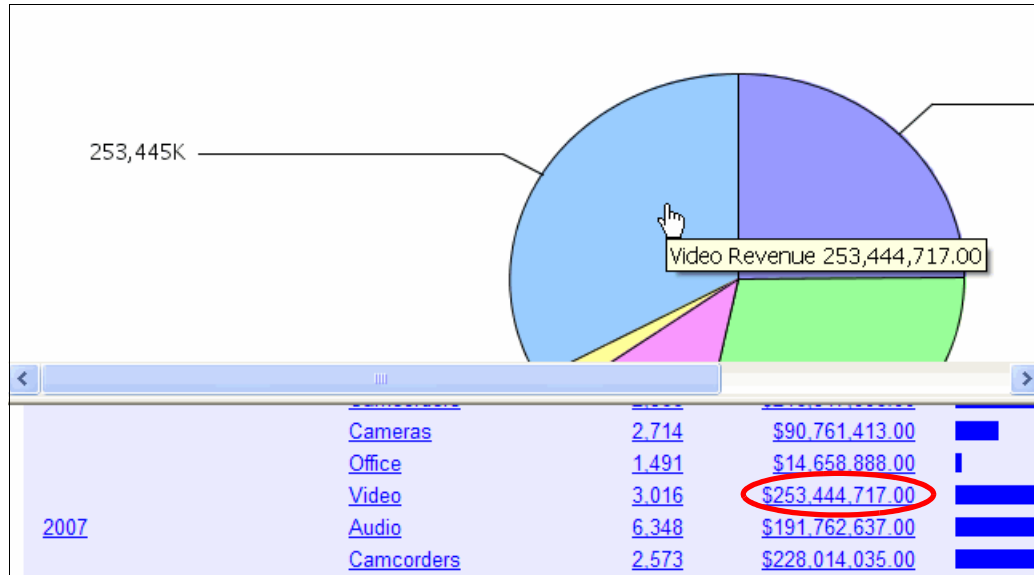


Figure 10-35 Graph drill down

Drilling down on the graph generates the same report as though you drilled down on the Revenue field in the original report.

28. Experiment with the various options for drilling down and filtering your report. When you are finished, click **Reset** to restore the report to its original values.

Tip: If our report has multiple measures that can be graphed, you can graph them on the same chart. You can select each measure for graphing and choose a different graph type. For example, if our report contains profit, you can superimpose a line graph on a bar graph with the line representing the profit and the bar representing revenue.

10.4.2 Manipulating the data using the OLAP Control Panel

Up to this point, we have manipulated the OLAP report through the Selection pane at the top of the report and directly through hyperlinks in the report itself. Now we use the third method for interacting with the data, which is the OLAP Control Panel.

1. Click the **OLAP** button at the bottom of the Selections pane to open the OLAP Control Panel. Alternatively, you can click the blue square next to Product Type.

Tip: If your OLAP Control Panel shows the selection criteria pane, click **OK** to return to the view shown in Figure 10-36.

Although the most frequently used functions are available directly from an OLAP report, the Selections pane, or both, several functions can be performed only from the OLAP Control Panel. The OLAP Control Panel can do everything that can be done from the OLAP report and the Selections pane and more.

The following unique OLAP functions, among others, can only be performed from the OLAP Control Panel:

- Selection of a Top or Bottom n rows report
- Ability to group numeric data by percentile (or decile, quartile, and so on)
- Advanced row selection criteria
- Stacking of multiple measures vertically in a report as opposed to horizontally

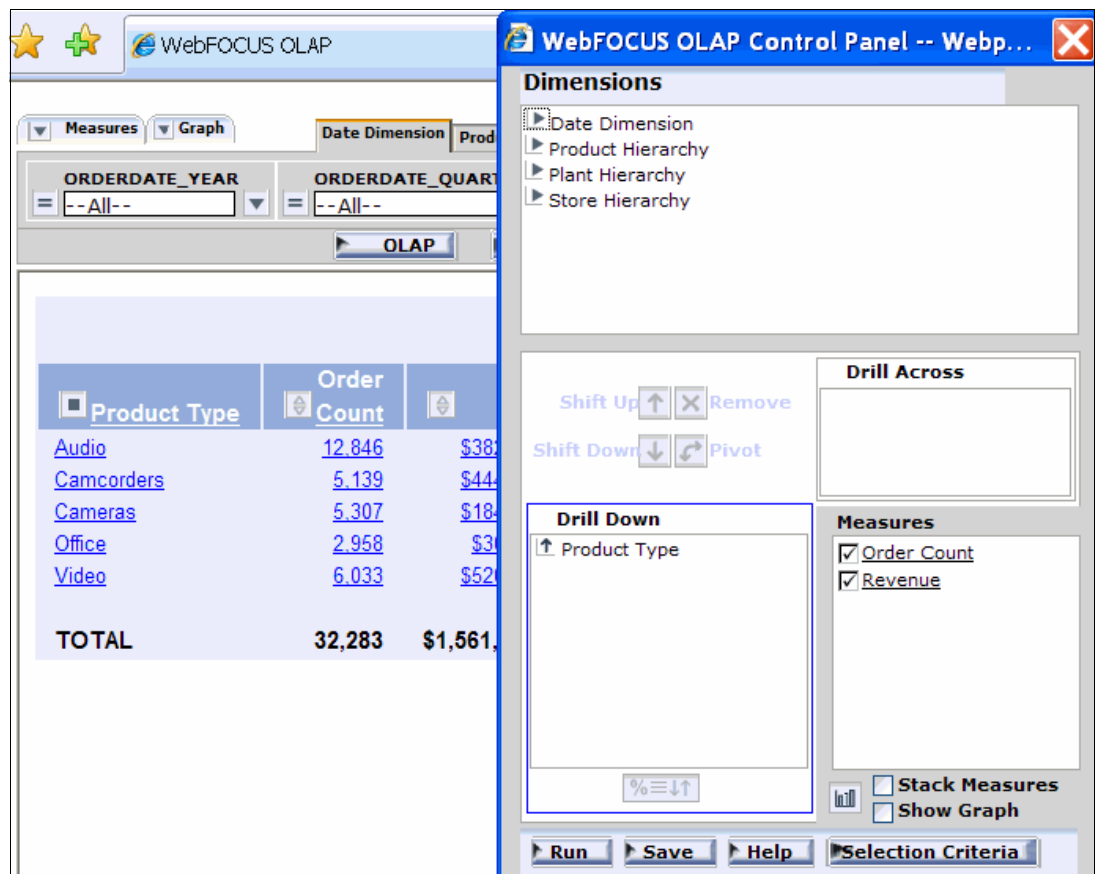


Figure 10-36 OLAP Control Panel

2. We want to see our data in columns by quarter:
 - a. Expand **Date Dimension**.
 - b. Click the **Drill Across** box. When selected, it is surrounded by a blue rectangle.
 - c. Click the **ORDERDATE_QUARTER** field. It should appear in the Drill Across box. See Figure 10-37.

Tip: If Quarter appears in the Drill Down box by accident, simply highlight it and use the Pivot button to pivot Quarter from rows to columns.

- d. Click **Run**.

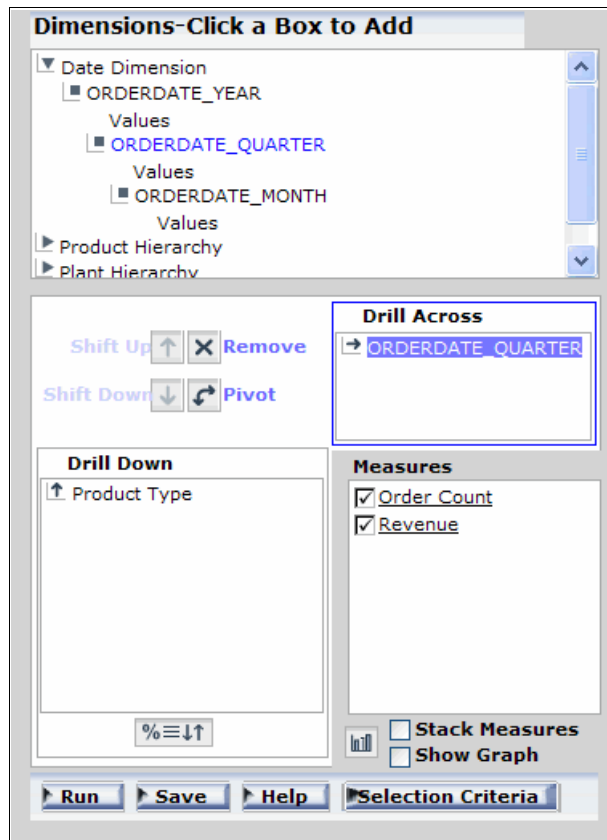


Figure 10-37 Specifying data in columns by quarter

e. Double-click **Quarter Q1** to drill down to month (Figure 10-38).

ORDERDATE_QUARTER						
	Q1		Q2		Q3	
Product Type	Order Count	Revenue	Order Count	Revenue	Order Count	Revenue
Audio	3,557	\$132,114,274.00	4,385	\$109,641,290.00	2,754	\$83,278,239.00
Camcorders	1,282	\$132,443,031.00	1,789	\$136,107,764.00	1,115	\$100,759,399.00
Cameras	1,393	\$57,319,083.00	1,830	\$55,163,758.00	1,152	\$41,902,533.00
Office	812	\$10,159,573.00	994	\$8,643,048.00	673	\$6,659,121.00
Video						
ORDERDATE_MONTH						
	01		02		03	
Product Type	Order Count	Revenue	Order Count	Revenue	Order Count	Revenue
Audio	487	\$18,674,656.00	1,360	\$53,845,383.00	1,710	
Camcorders	142	\$14,687,978.00	481	\$53,339,449.00	659	
Cameras	161	\$6,066,787.00	539	\$23,208,633.00	693	

Figure 10-38 Cross-tab drill down

- Often when you have multiple measures and sort across columns, the report can become wide and unwieldy. Open the OLAP Control Panel and select **Stack Measures** (Figure 10-39). Click **Run**.

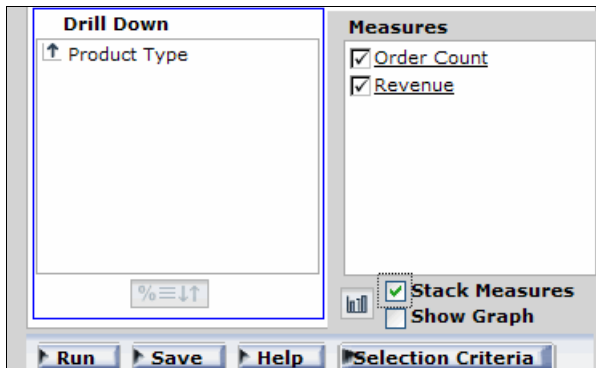


Figure 10-39 Stack Measures option

There is a potential problem with this report as it stands right now. If the Orders table contains data for multiple years, we now have the sales for every Q1 added up together (Figure 10-40). Although this is technically accurate, it probably is not what the user wants to see.

		ORDERDATE_MONTH		
		01	02	03
Product Type				
Audio	Order Count	487	1,360	
	Revenue	\$18,674,656.00	\$53,845,383.00	\$59,591,111.00
Camcorders	Order Count	142	481	
	Revenue	\$14,687,978.00	\$53,339,449.00	\$64,411,111.00
Cameras	Order Count	161	539	

Figure 10-40 Stacked measures report

- From the OLAP Control Panel, expand **Date Dimension** and select **Values** for ORDERDATE_YEAR (Figure 10-41).

Note: For most dimensions, when you click Values, you see the list of values from your table. The date components are prefilled in with general dates. If you expand ORDERDATE_MONTH, you see that the actual month names, that is June, are filled in for ease of use for the user.

- In the window that opens (lower half of Figure 10-41), select the **Range** box. This is not appropriate for our data that contains only records from 2006 and 2007, but it is interesting to see the capability. Deselect **Range**.

For Year, select **2007** and click **Add**.

Dimensions-Click a Box to Add

Date Dimension

ORDERDATE_YEAR

Values

ORDERDATE_QUARTER

Values

ORDERDATE_MONTH

Values

Product Hierarchy

Plant Hierarchy

ORDERDATE_YEAR

☐ Range

Year

2007

Selections

2007

Add

Delete

Run

Save

Help

Ok

Cancel

Figure 10-41 Filtering an OLAP report

6. At this point, you can click Run. However, for demonstration purposes, click **OK** to see the full Selection Criteria pane.
7. In the Selection Criteria pane (Figure 10-42), notice that Q1 is prefilled in for ORDERDATE QUARTER, because we drilled down and selected Q1 prior to opening the control panel.

If you use the scroll bar on the right, you see all the various dimensions and their individual elements or columns that were available through the tabs in the Selection pane. You can continue to filter your data here.

If you want to filter on Product Type, you have two choices. You can click the down arrow next to Product Type or you can click the Select button. In both cases, you see a list of all the unique values for Product Type. Using the Select button takes you to a second window. This is easier to use when you are selecting multiple values.

Again, when filtering, you can modify the operand for each filtered column. Here, we do not filter on Product Type.

Click **Run**.

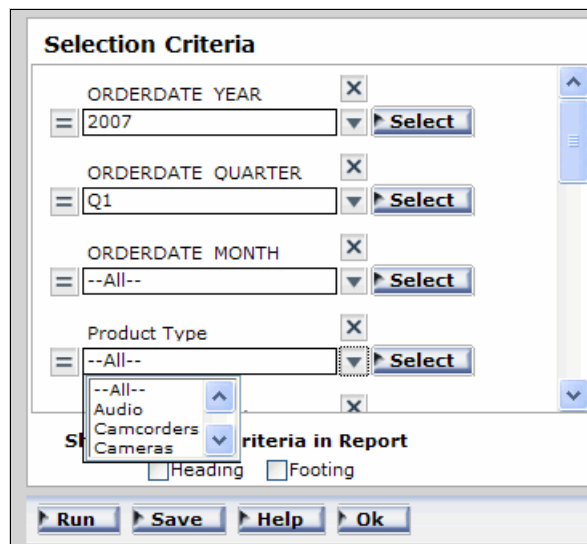


Figure 10-42 OLAP filtering

Figure 10-43 shows the output of running the report.

Note: Best practice states that you should always display any filters in effect on a report. DB2 Web Query has the capability from the OLAP Control Panel to document your filters at the beginning or end of your report.

Measures

Graph

Date Dimension

Product Hierarchy

Plant Hierarchy

Store Hierarchy

ORDERDATE_YEAR

2007

ORDERDATE_QUARTER

Q1

ORDERDATE_MONTH

--All--

OLAP

Run

Reset

Save

Help

ORDERDATE_MONTH

01

02

03

Product Type

Audio	Order Count	245	705	832
	Revenue	\$8,891,604.00	\$29,349,235.00	\$28,483,882.00
Camcorders	Order Count	71	240	318
	Revenue	\$7,870,962.00	\$28,685,767.00	\$31,846,307.00
Cameras	Order Count	74	265	330
	Revenue	\$2,682,543.00	\$11,867,208.00	\$13,453,754.00
Office	Order Count	48	167	201
	Revenue	\$751,728.00	\$2,303,002.00	\$2,580,783.00
Video	Order Count	106	293	385
	Revenue	\$12,924,891.00	\$37,653,669.00	\$34,617,427.00
TOTAL	Order,Count	544	1,670	2,066
	Revenue	\$33,121,728.00	\$109,858,881.00	\$110,982,153.00

Figure 10-43 2007 only report

- If you are still on the Selection pane in the OLAP Control Panel, click **OK** to return to the initial layout. Notice that you can click Selection Criteria in the bottom right of the Control Panel to return directly to the Selection Criteria layout.

9. In the Measures pane (Figure 10-44), select the **Revenue** check box. Then click the **Revenue** name.

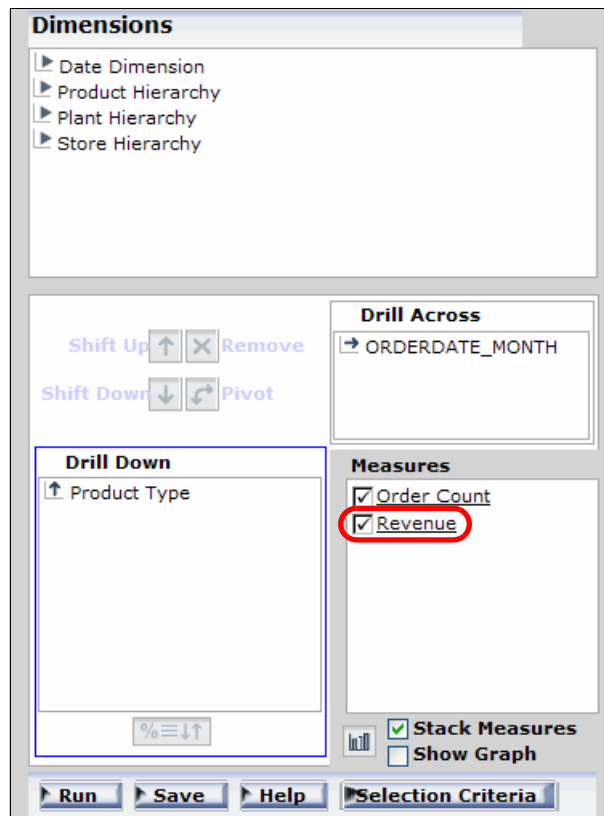


Figure 10-44 Measures pane

10. In the Revenue pane (Figure 10-45 on page 231), if we want to see a Top 3 analysis report:

- Select the **Sort** check box.
- Select **High to Low**.
- Select the **Rank (1, 2, 3 ...)** check box.
- For Highest, select **3**.
- For Measure Calculations, keep the default of **None**.

Notice that if you expand Measure Calculations, you can ask to see the Revenue column in multiple ways. For example, you can ask to see each Revenue as a percentage of the product type revenue in that quarter (Row Percent). You can ask to see the average revenue per order and much more.

Tip: If you want to see a report that shows your bottom Products by revenue, you sort from low to high and then rank and select your top three.

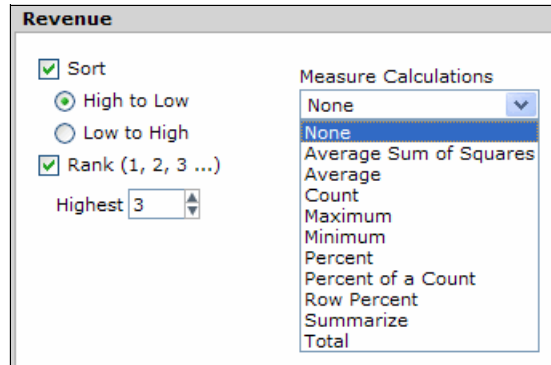


Figure 10-45 Top 3 OLAP report

10.4.3 Grouping data into percentiles

The reports and data used in these tutorials do not lend themselves to grouping into tiles (percentiles, deciles, quartiles, and so on), but you should be aware that this capability exists.

The example described in this section starts with a report that has one record for each sales person showing their revenue attainment. The company wants to reward the top 10% of their sales people. For example, if the company has 200 sales people, it wants to know who the top 20 people are. If the twentieth and twenty-first sales people are tied, the twenty-first sales person moves into the top ten percent as well.

From the OLAP Control Panel (Figure 10-46), we highlight the revenue field LineTotal1 and click the Sort button.

Sales reps revenue	
"Line Total1"	"SalesRep"
135294247.00	Robin DeWitte
130778087.00	Web
50379924.00	Arild Kristensen
49906101.00	Bjorn Danielson
46142505.00	Marc Wacongne
44347392.00	Tsutomo Nakanishi
43419919.00	Barry Phillips
36016255.00	Patrick Churchville
36006767.00	Ge Kapteijn
35714713.00	Betty Hong
32470686.00	Elisabeth Rochon-Birac
28482733.00	Martin Nissim
28018061.00	Roger Iddles
26856448.00	Wendy Hutson
26404569.00	Jong Soo Kim
26361768.00	Theresa Wellman
25037413.00	Marc Guille
24822455.00	Ellen Roccisano

Figure 10-46 Starting report showing sales people and their attainment

In the next pane (left in Figure 10-47), we choose a high to low sort sequence and then select the Tiles tab. On the Tiles tab (right in Figure 10-47), we select the options to tile the report in groups of 10 so that we can see our revenue numbers evenly split into ten groups. We want the new column with the group number to be called *Decile*. If we only want to see the top 10%, we restrict the report to the first tile only.

If we want to see the top 5% of the sales people, we can ask the system to split the sales people into 20 groups and then we choose the top group. We can also split the sales people into 100 groups with 1% in each group. To get the top 5%, we choose groups 1 - 5.

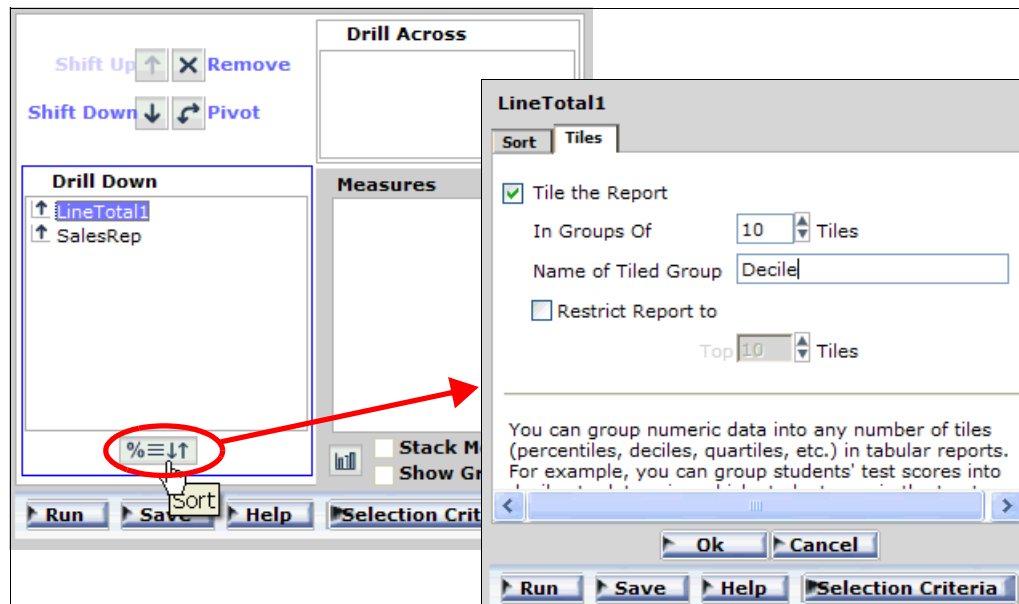


Figure 10-47 Sorting *LineTotal1* into groups of 10



HTML Layout Painter

HTML Layout Painter is a component of IBM Developer Workbench. Prior to starting this set of tutorials, you must have completed Chapter 9, “Developer Workbench” on page 193.

HTML Layout Painter is specifically designed, as the name suggests, to design HTML pages. HTML Layout Painter does not create reports or graphs, but rather references existing ones. You can create compound reports with background style sheets and add company logos and other standard HTML components. The design of the HTML page can be set to match your company’s intranet standards.

Both HTML Layout Painter and Power Painter can create compound reports. Power Painter is architected for PDF output and has additional controls specific to large PDF reports. It can also create other types of output such as HTML pages. HTML Layout Painter is architected for HTML pages; it can only create HTML pages. It cannot create a PDF report. The products both have different user interfaces. You have already tried Power Painter. In this tutorial, you try HTML Layout Painter. Choose the option that fits your company’s needs the best.

In this tutorial, we design two different HTML pages. The first is a compound report with a single input parameter. This style of report can be used as the basis for key performance indicator (KPI) dashboards where the user might request the KPIs for a specific country, product type, and so on. This type of report is often designed by IT and deployed to senior management and executives within a company.

The second report that we design is a single, highly parameterized active report. Again, this report is typically developed by IT but, in this case, is often deployed to business analysts or disconnected users who want to highly segment the data they are analyzing. This tutorial assumes that you have completed the Active Reports tutorial (Chapter 8, “Active Reports” on page 175) and understand those concepts. Here we take an existing active report and initiate it from an HTML page that contains a large number of input parameters in a user friendly manner.

11.1 Compound parameterized report (HTML1_KPI)

In the first exercise, using two graphs and one report created in previous exercises, we add a common parameter and bring them together on one HTML page for the user. Figure 11-1 shows the final output page.

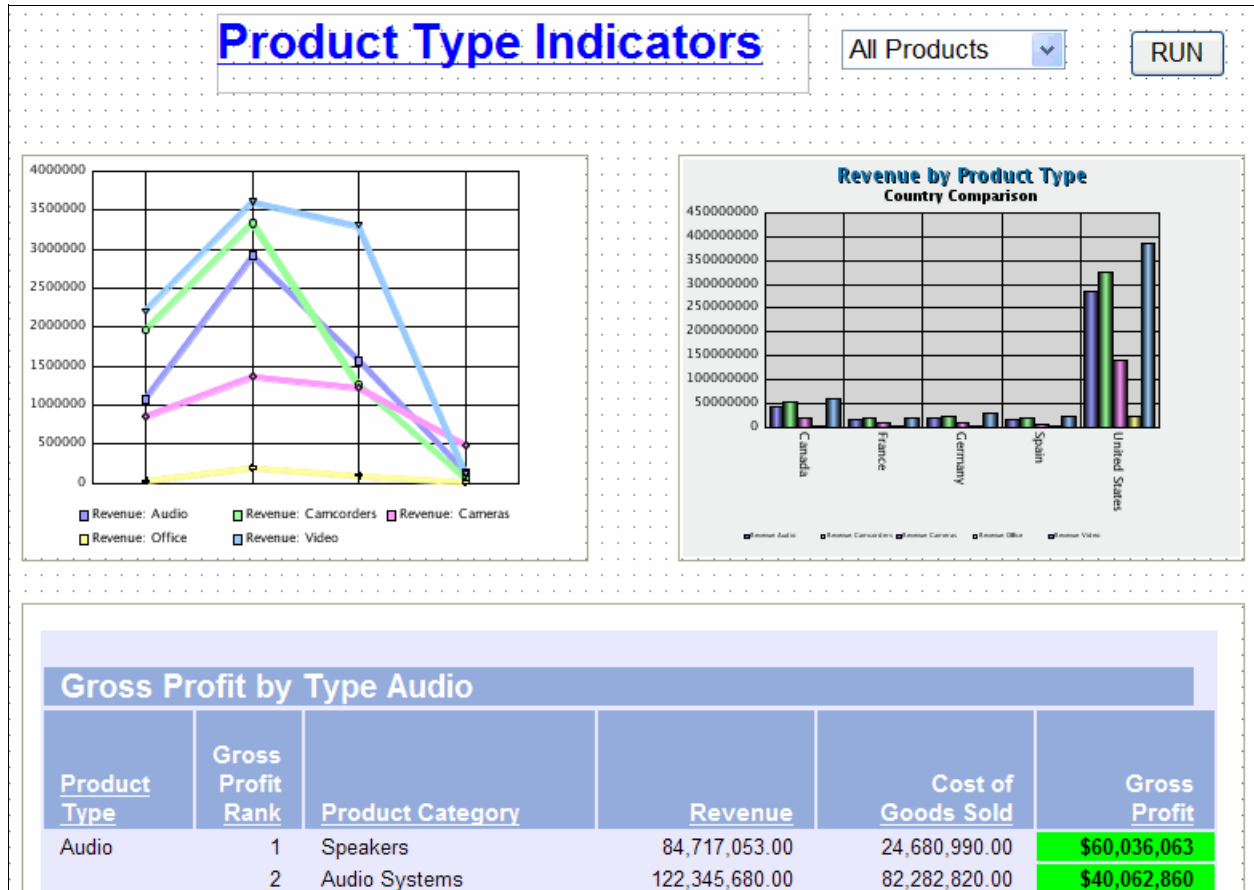


Figure 11-1 Compound report

11.1.1 Adding parameters to existing reports

In this tutorial, we do all our work from Developer Workbench. You can run Report Assistant and Graph Assistant from either Developer Workbench or the DB2 Web Query home page.

In Chapter 9, "Developer Workbench" on page 193, we introduce the two main functional areas of Developer Workbench. The first leg of the explorer tree expands through Data Servers to allow you to work with your metadata. The second branch expands down through Managed Reporting to allow you to work with the folder in your domains.

In this section, we must add product type as a parameter to the three reports that we want on our KPI page. We modify the existing reports to accept this parameter and then save them under new names.

Generic term report: The generic term report is used for both text reports and graphs. It is also used to refer to the completed HTML page. In the first exercise, the completed HTML page is a compound report.

1. Open Developer Workbench. You see WebFOCUS Environments under Developer Studio Desktop. You may or may not see Windows Desktop. This is an optional setting described in Chapter 9, “Developer Workbench” on page 193.
2. Expand **WebFOCUS Environments** → *your environment* → **Managed Reporting** → **Domains** → **Common Domain** → **Reports** → **Tutorials**.
3. In your Tutorials folder, navigate to the graph **GA3_Trend**, right-click, and select **Copy** (Figure 11-2).

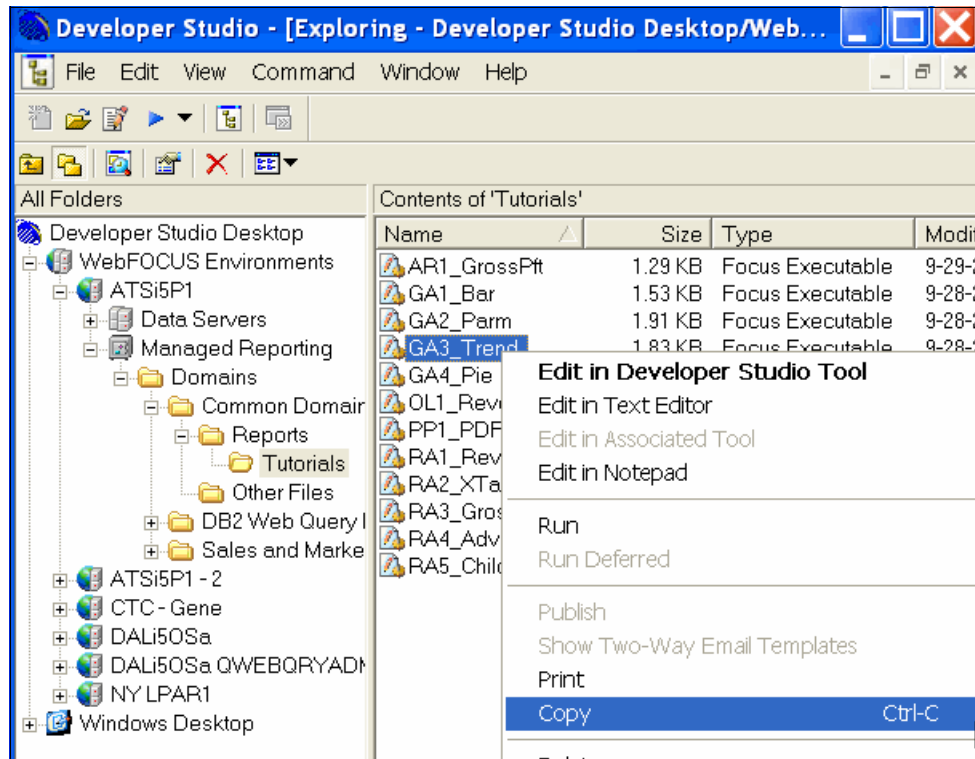


Figure 11-2 Developer Workbench reports

4. Keeping your cursor in the same pane (the list of your reports), right-click and select **Paste**. You now have a report titled *GA3_Trend1* in the Tutorials folder.

Restriction: When accessed by Developer Studio, Report Assistant does not currently support the Save As capability. This is a known requirement. As a workaround, we use Copy and Paste to place the report into the same directory.

5. Right-click your new report **GA3_Trend1** and select **Rename**. Give the report the name HTML1_KPI_LINE.

Tip: Even though you rename the report in the window to HTML1_KPI_LINE, if you look in the integrated file system or look at the properties of the new report, you see that the object name is still *ga3_trend1.fex*. You have changed only the display title of the report, not the physical name on disk.

- Right-click **HTML1_KPI_LINE** and select **Edit in Developer Studio Tool** (Figure 11-3).

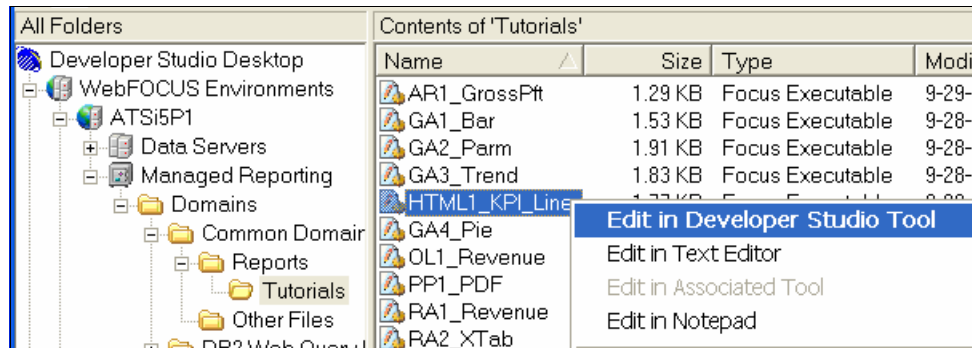


Figure 11-3 Selecting the Edit in Developer Studio Tool option

- DB2 Web Query knows which tool was used to create your report. Selecting the default option opens the report in either Report Assistant or Graph Assistant, whichever is appropriate. Power Painter is not directly supported in Developer Workbench. In this example, Graph Assistant opens.
- In Graph Assistant, choose the **Selection criteria** tab.
- On the Selection criteria tab (Figure 11-4):
 - Delete all existing selection criteria.
 - Drag **PRODUCTTYPE** to the Selection pane.
 - Click **Select values**.
 - In the EQ Values window, select **Parameter** and then click the **Auto Prompt** button. Accept all the defaults for PRODUCTTYPE prompting. Click **OK**.
 - Back on the EQ Values window, click **OK**.

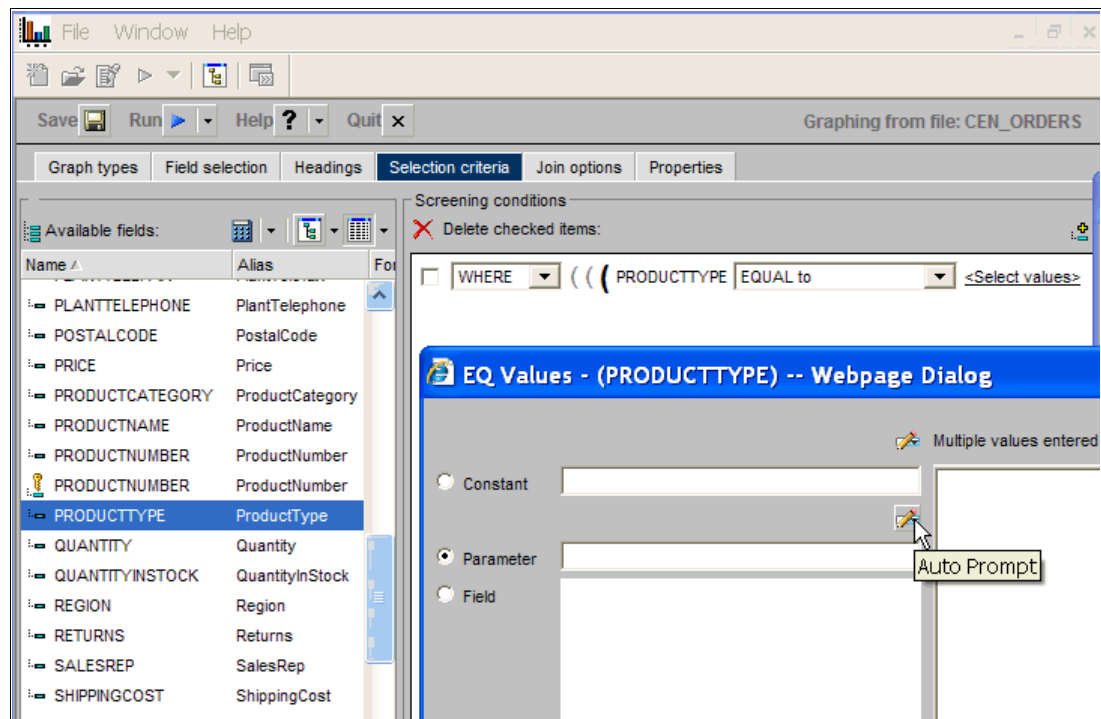


Figure 11-4 Selection criteria and parameter prompting

Notice in the Selection pane (Figure 11-5) that the Select values link has changed.

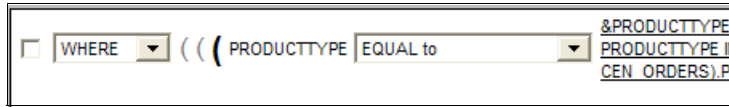


Figure 11-5 Completed selection criteria

10. Remove the months from the X axis labels to free more room in the small output frame for the graph itself.

- a. Click the **Properties** tab and the **X axis** subtab (Figure 11-6).
- b. On the X axis subtab, clear the **Show axis labels** check box.

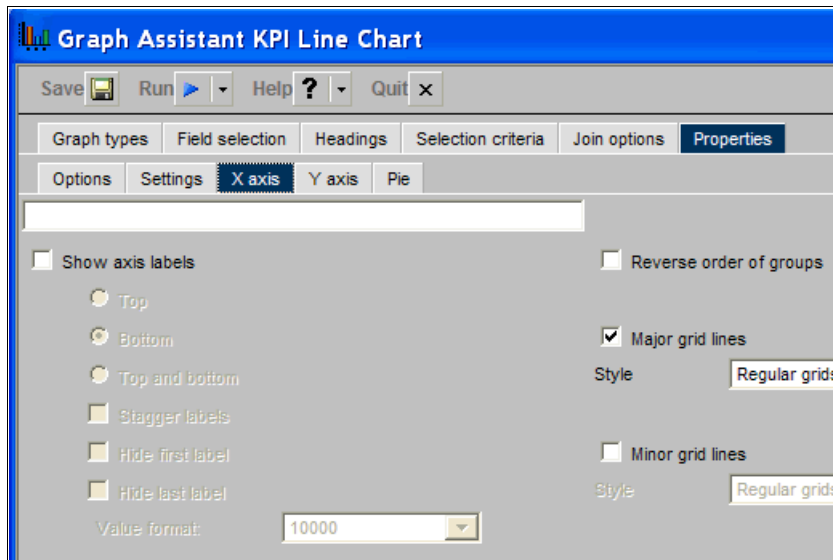


Figure 11-6 Removing the X axis labels

- c. Click the **Options** subtab (Figure 11-7).
- d. From experience, we know that the frame for our graph is about 350 pixels wide by 250 pixels high. This can be an iterative process. You might need to design your HTML page, see how much room you have for your graph, and then return to the output options for the graph to ensure that it will fit in the designated space.

Set the output size to **350** pixels wide (Horizontal) by **250** pixels high (Vertical).

- e. Save your graph.

The screenshot shows a software window titled 'Graphing from file'. It has a menu bar with 'Save', 'Run', 'Help', and 'Quit'. Below the menu bar are several tabs: 'Graph types', 'Field selection', 'Headings', 'Selection criteria', 'Join options', and 'Properties'. The 'Properties' tab is active, and within it, the 'Options' subtab is selected. The 'Options' subtab contains two main sections: 'Graph size' and 'Output'. In the 'Graph size' section, there are three radio buttons: 'Use WebFOCUS default.', 'Let application choose size.', and 'Custom size.'. The 'Custom size.' option is selected. Below these radio buttons are two input fields: 'Horizontal' with the value '350' and 'Vertical' with the value '250'. In the 'Output' section, there is a dropdown menu for 'Output to' set to 'Screen', a text field for 'Name', and a dropdown menu for 'Format' set to 'PNG'. At the bottom of the window, there are two more sections: 'Background color' and 'Frame color', each with a color selection button, and a section for 'Limit the number of read operations performed during' with a text input field.

Figure 11-7 Properties tab

11. We repeat the previous steps for the existing bar chart GA1_Bar. If you need more detailed assistance, refer to the previous steps and figures.
 - a. Copy and paste the **GA1_Bar** and rename it **HTML1_KPI_Bar**.
 - b. Add a **PRODUCTTYPE** parameter, accepting the auto prompting defaults. Your report should have only one parameter.
 - c. Set the output size to **350** pixels wide by **250** pixels high.
 - d. Click the **Field selection** tab (Figure 11-8).
 - e. Click **Show - Field options**. To save space for our graph, for the X axis field label field, delete **Country**.
 - f. Save your graph.

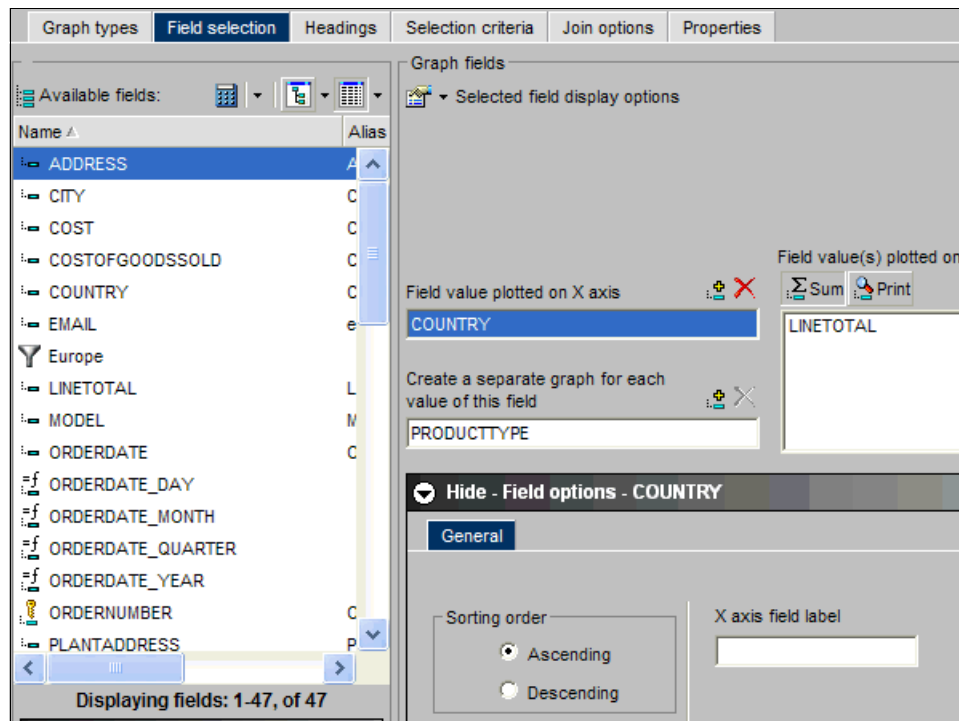


Figure 11-8 X axis field label deleted

11.1.2 Designing the layout for the HTML report

In this section, we explain how to design the layout for an HTML report:

1. Open Developer Workbench.
2. Expand **WebFOCUS Environments** → *your environment* → **Managed Reporting** → **Domains** → **Common Domain** → **Reports** → **Tutorials**.

3. Right-click the **Tutorials** folder and select **New** → **HTML File** (Figure 11-9).

Tip: To create a text report or graph, select **New** → **Procedure**. You then see a window that allows you to choose between using Report Assistant, Graph Assistant, and the SQL Wizard to create your new report.

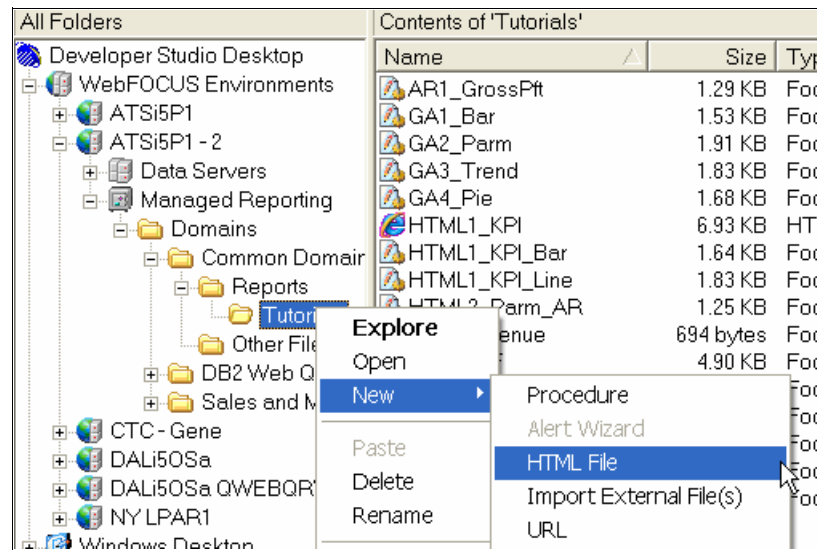


Figure 11-9 New HTML File

4. Call your new report HTML1_KPI and click **Create**.

In the next subsections, we explain how to include a line chart, a bar graph, and the gross profit report. Figure 11-10 shows the final layout design of our compound report. The sections that follow refer to this figure unless noted otherwise.

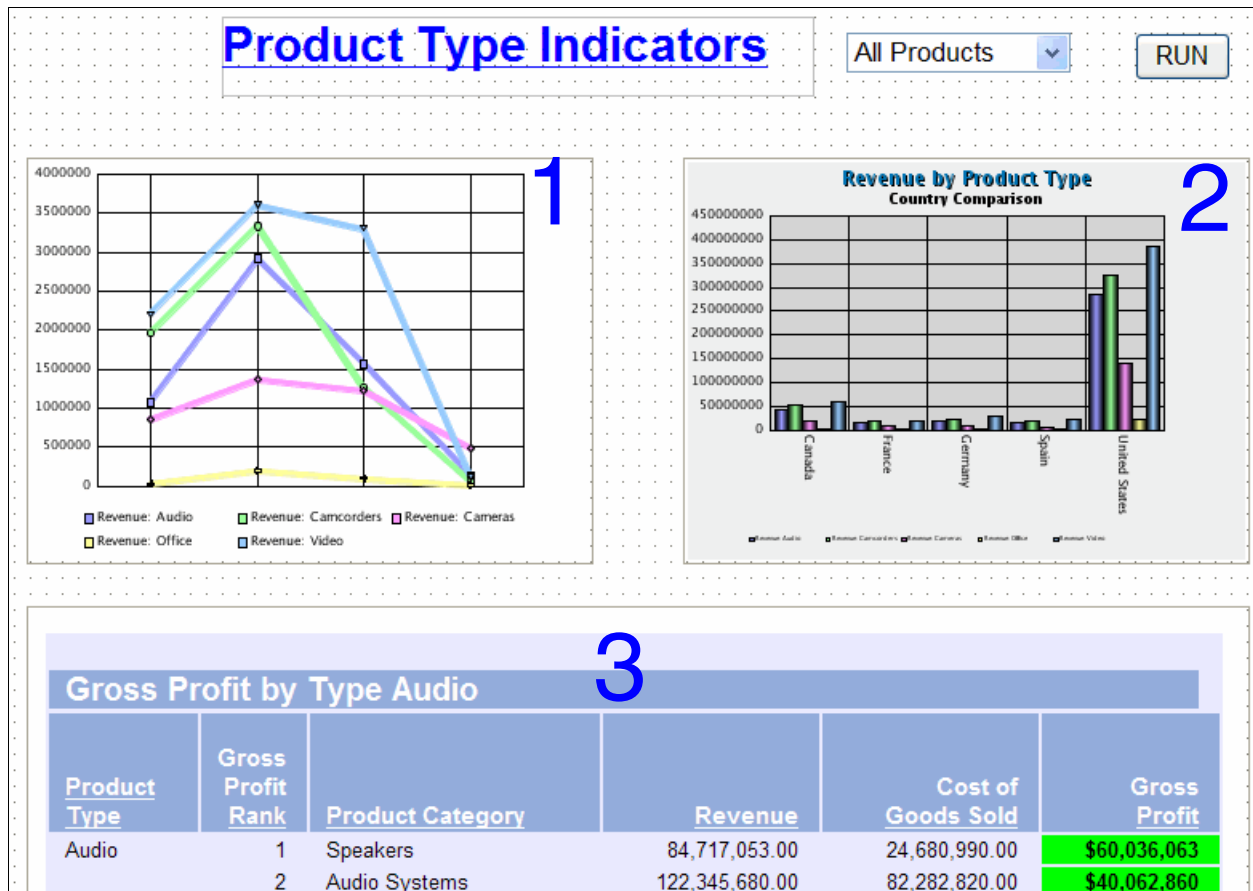


Figure 11-10 Design layout for HTML report

Including a line chart

To include a line chart in the HTML report:

- Click the **graph** icon from the toolbar (circled in Figure 11-11).

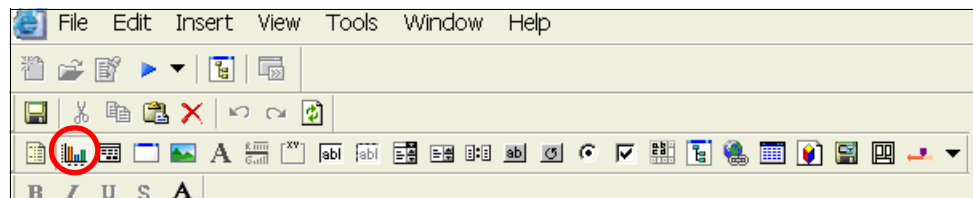


Figure 11-11 HTML Layout toolbar

- Draw a box similar to the box in the upper left corner labeled “1” in Figure 11-10. Look at the properties box in the right pane. Your box should be 250 pixels high and 350 pixels wide. This is the size we designed our graph for.

7. Right-click **box 1** and select **Reference existing procedure** (Figure 11-12). This not only shows how the graph will look in this frame, but it also references any parameters that are used by the graph.

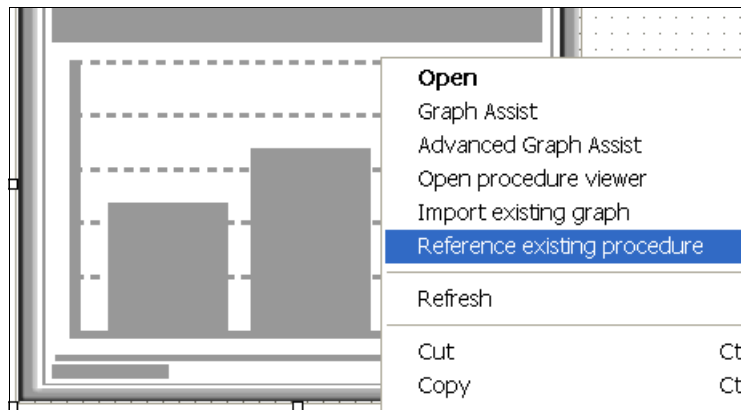


Figure 11-12 Reference existing procedure

8. Open your **HTML1_KPI_Line** file in the Tutorials folder. Notice that the actual file name is displayed (Figure 11-13).

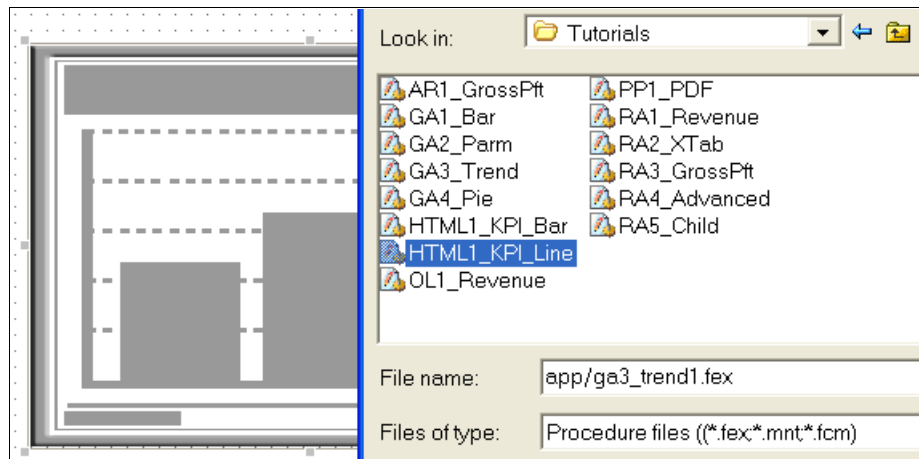
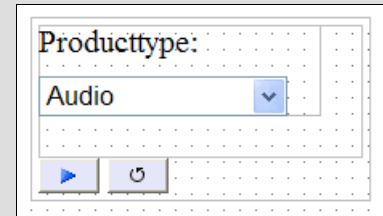


Figure 11-13 Procedure being referenced

9. In the New Parameters window (Figure 11-14), the system presents a list of the parameters used in HTML1_KPI_Line.
 - a. For Control Type, accept the default of **Drop-down list**.
 - b. For Parameter grouping options, select **Do not create a form**.

New form element: If you accept the default New form element, the system creates a panel that contains your parameter label, the drop-down box, a run button and a refresh button, like the following example. We said no to this because we want to be able to move each of these elements separately on our page.



- c. Click **OK**.

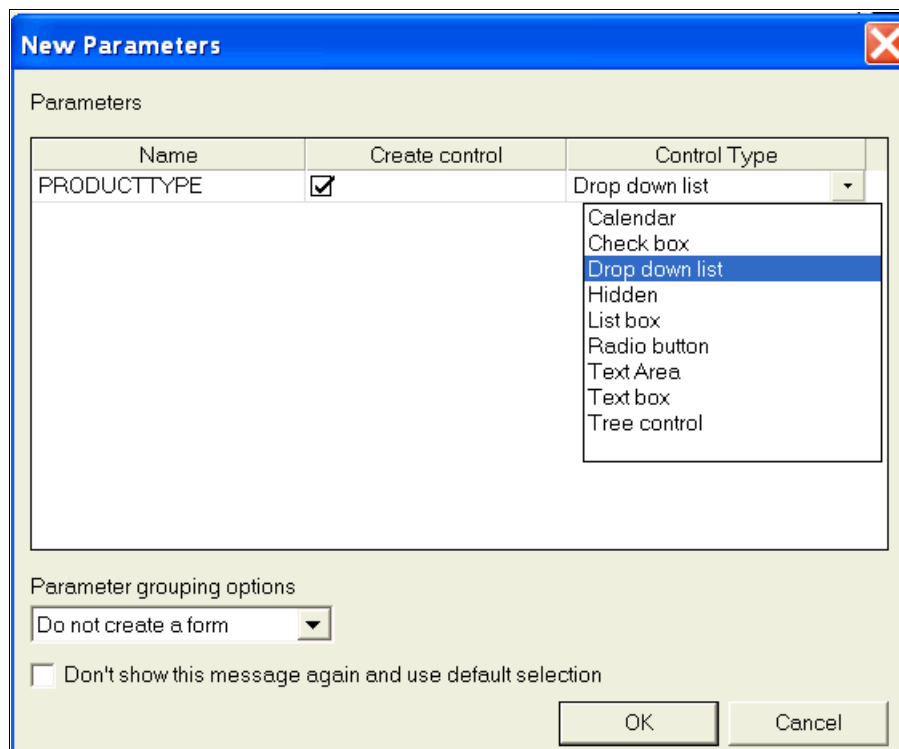


Figure 11-14 Defining a parameter

10. Delete the label **PRODUCTTYPE**. We reference Product Type in our overall report heading.
11. Move the drop-down list and the Run button as shown in Figure 11-10 on page 241. Your Run button looks slightly different at this stage.

12. Select the line graph. In the Properties pane on the right, change Name to LineGraph (Figure 11-15).

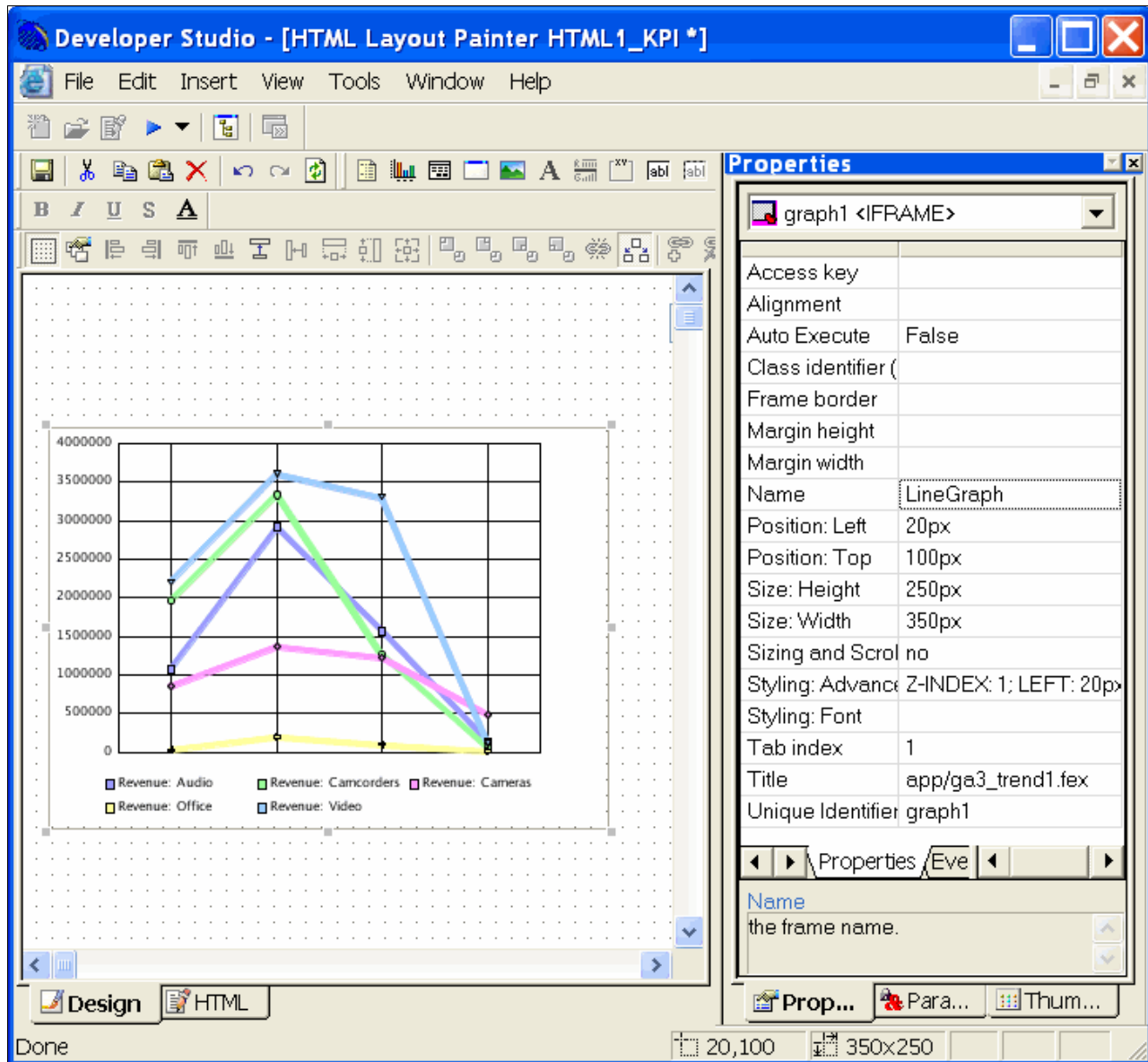


Figure 11-15 Setting the name property

13. Click the drop-down control box in the report (circled in Figure 11-16). Click the **Parameters** tab in the lower right corner. Select the **Add "ALL" option** and specify to display All Products.

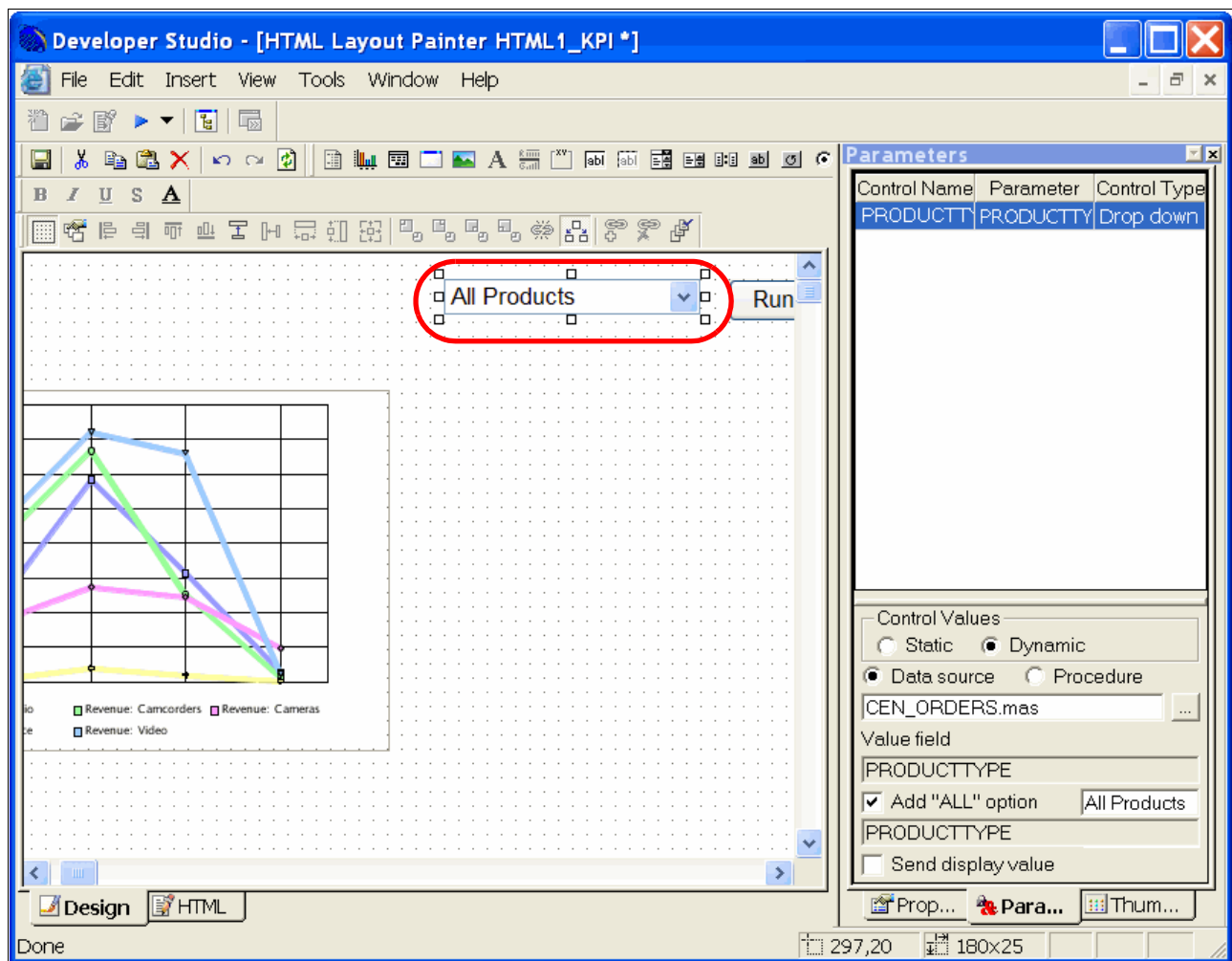


Figure 11-16 Adding the ALL option to a parameter

Including a bar chart

To include a bar chart in the HTML report:

14. Click the **graph** icon.
15. Draw a box similar to the box in the upper right corner in Figure 11-10 on page 241 labeled "2".
16. Select the **Properties** tab at the bottom of the right pane.
17. Right-click the **new graph frame** and select **Reference existing procedure**.
18. Select **HTML1_KPI_Bar**.
19. To make the two graph frames the same size, you can use the Properties pane and manually size the frames to both be 250 x 350.
We use the alternative to select both frame 2 and frame 1 using the Ctrl key and click the **Make same size** icon (⌘). Notice that it is the *last* element selected that controls the final size of all the selected elements.

20. Again select **frame 2** and then **frame 1** and click the **Align top** icon ().
21. In the Properties pane on the right, change the name to BarGraph.

Including the gross profit report (RA4_Advanced)

To include the gross profit report:

22. Select the **report** icon (Figure 11-17).

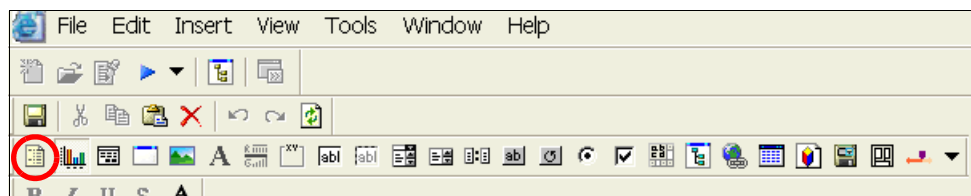


Figure 11-17 HTML Layout Painter toolbar

23. Draw a box similar to the box in the upper right corner in Figure 11-10 on page 241 labeled “3”.
24. Right-click the **frame** and point to the **RA4_Advanced** report.
25. In the Properties pane on the right, set Size and Scrolling to **No**. This automatically adjusts the size of your frame so that your report fits in it and you do not have to scroll within the frame. Change the Name to GrossPftReport.

HTML page options

To make this page look a little cleaner, we remove the frames around each of the reports:

26. Highlight each of the three reports.
27. From the Properties pane on the right, set Frame border to **No**.
28. To have the all three reports in our compound HTML page automatically run when the page is opened, highlight each of the three reports and from the Properties pane on the right, set Auto Execute to **True**.
29. Give the report a title.
 - a. Select the text icon (Figure 11-18).

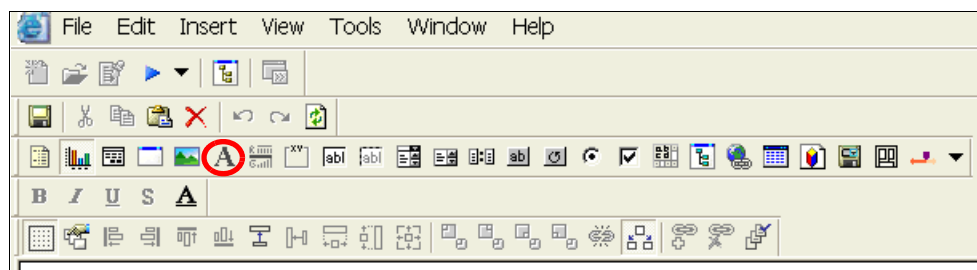


Figure 11-18 Adding text to an HTML page

- b. Draw a box in the pane where you want the report title to be.
 - c. Enter the text Product Type Indicators.
30. Click your text box so that it is surrounded by a straight line frame and not slashes.
31. In the Properties pane on the right, click the ellipsis button (...) next to Styling: Font.

32. Define the styling for your heading by selecting the following options: **arial**, **18pt**, **bold**, **underlined**, and **blue**.

Executing the Run button

Finally, we tell the system which reports need to execute when the Run button is clicked:

33. Right-click the **Run** button and choose **Hyperlink properties**.

34. In the Hyperlink Properties window, change the display text to RUN and appropriately size the RUN button.

35. Click the **add** icon (🔍) on the right.

36. The Run button is initially set to execute the original report that was used to import the parameter. We need to tell it to also execute our other two reports when the user clicks Run.

- To add the second graph, for Action, select **External procedure**.
- For Source, click the ... button, and navigate to and select **HTML1_KPI_Bar**.
- For Target Name, select **BarGraph** to indicate which frame the report should run in.
- Repeat the process and select the external procedure RA4_Advanced and point it to the frame GrossPftReport.

Your Hyperlink Properties window should look like the example shown in Figure 11-19.

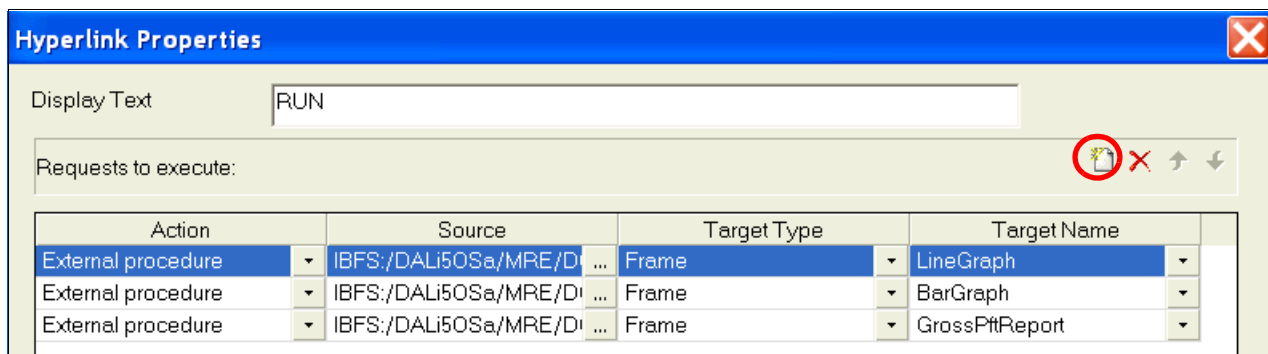


Figure 11-19 Hyperlink Properties

37. Save your report and then run it by clicking the Run button (circled in Figure 11-20).

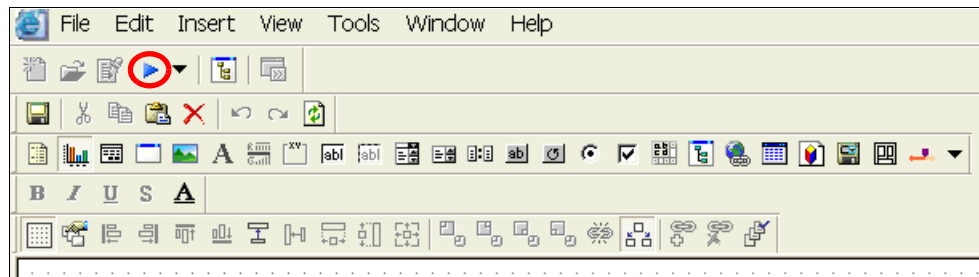


Figure 11-20 Run button

Tip: You can return to the DB2 Web Query home page and run this dashboard. The easiest method is for a developer to stay within Developer Workbench and test from there. Do not forget that the page should be designed small enough to be viewed within the context of the DB2 Web Query home page.

Your dashboard or KPI should look like the example in Figure 11-21.

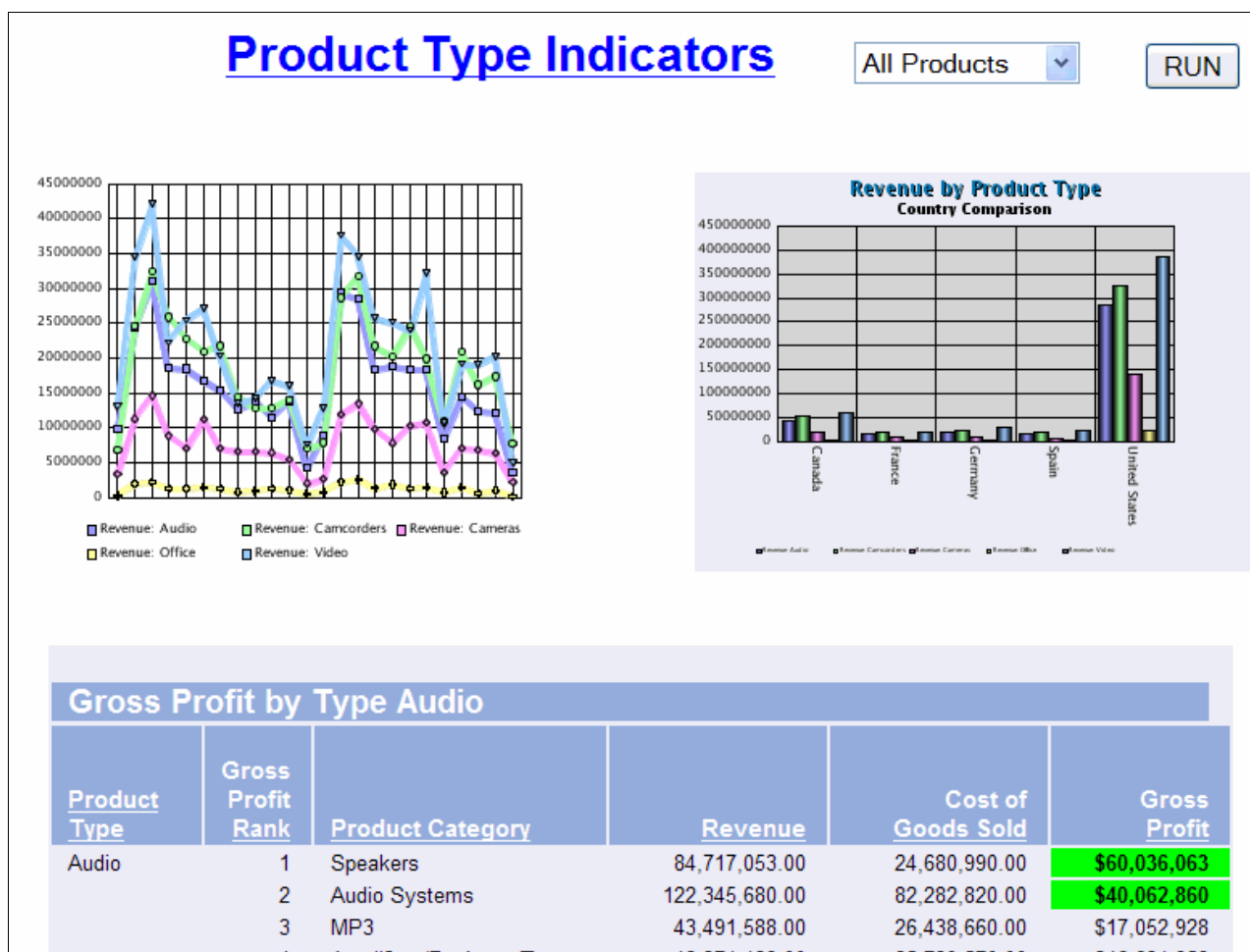


Figure 11-21 Completed compound report

11.2 Highly parameterized report (HTML2_Parm)

In this exercise, using an active report that we created previously, we add multiple parameters to it. This allows a user to segment the data that they want to analyze in different ways. Although we base this tutorial on an active report, we can invoke any report or graph. This highly parameterized technique is effective for subsetting the data that a user wants to download to Excel.

This exercise uses the Active Report AR1_GrossPft, which is shown in Figure 11-22 on page 249. You might want to quickly run this report and make sure that it looks like the example in the figure.

Tip: If your active report is in the accordion format, open the report, go to **Report options**, choose **Active Report Styling** and select **Tabular** for Report view. Save your report.

Gross Profit by Product Category				
Product Type ▼	Product Category ▼	Line Total Title ▼	Cost of Goods Sold ▼	Gross Profit ▼
Audio	Speakers	84,717,053.00	24,680,990.00	\$60,036,063
	Audio Systems	122,345,680.00	82,282,820.00	\$40,062,860
	MP3	43,491,588.00	26,438,660.00	\$17,052,928
	Amplifiers/PreAmps/Tuners	42,374,428.00	25,739,570.00	\$16,634,858
	CD Players and Recorders	53,847,459.00	37,838,460.00	\$16,008,999
	Receivers	35,907,113.00	22,998,000.00	\$12,909,113
	Subtotal for Audio	382,683,321.00	219,978,500.00	\$162,704,821
Camcorders	DVD Camcorders	379,376,637.00	300,373,350.00	\$79,003,287
	MiniDV Camcorders	51,539,451.00	34,128,360.00	\$17,411,091
	Digital8 Camcorders	13,614,953.00	6,512,600.00	\$7,102,353
Subtotal for Camcorders	444,531,041.00	341,014,310.00	\$103,516,731	
Cameras	Digital Cameras	184,103,667.00	133,328,830.00	\$50,774,837
Subtotal for Cameras	184,103,667.00	133,328,830.00	\$50,774,837	
Office	Organizers	11,712,495.00	4,957,305.00	\$6,755,190
	Handheld and PDA	18,533,190.00	14,067,420.00	\$4,465,770
Subtotal for Office	30,245,685.00	19,024,725.00	\$11,220,960	
Video	DVD	329,872,045.00	248,768,900.00	\$81,103,145
	TV	168,799,539.00	150,771,700.00	\$18,027,839
	VCR	21,688,621.00	16,270,950.00	\$5,417,671
Subtotal for Video	520,360,205.00	415,811,550.00	\$104,548,655	

Figure 11-22 AR1_GrossPft Active Report

To create the report:

1. Open Developer Workbench.
2. Expand **WebFOCUS Environments** → *your environment* → **Managed Reporting** → **Domains** → **Common Domain** → **Reports** → **Tutorials**.
3. In your Tutorials folder, navigate to the **AR1_GrossPft** report, right-click, and select **Copy**.
4. Keeping your cursor in the same pane (the list of your reports), right-click and select **Paste**. You now have a report titled *AR1_GrossPft* in the Tutorials folder.
5. Right-click **AR1_GrossPft** and select **Rename**. Give the report the name **HTML2_Parm_AR**.
6. Open the new report. In the right pane, right-click **HTML2_Parm_AR** and select **Edit in Developer Studio Tool** (Figure 11-23).

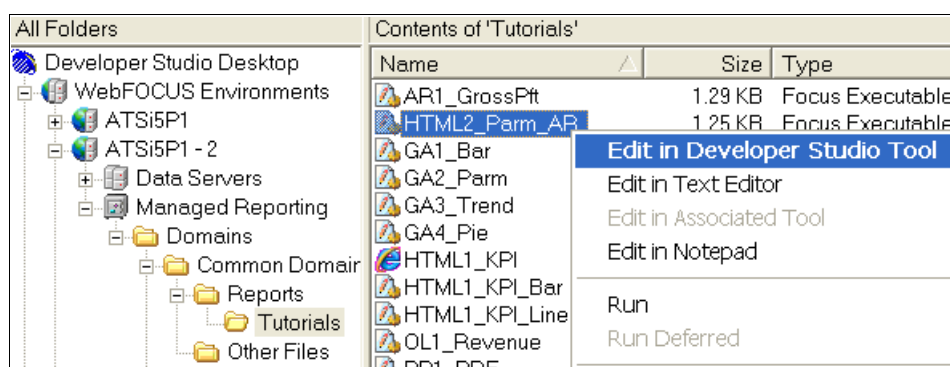


Figure 11-23 Opening the report in Developer Workbench

- Click the **Selection criteria** tab and enter, in sequence, the parameters shown in Table 11-1.

Based on this table, all the conditions are WHERE conditions. The table shows the field name to drag to the Selection criteria pane and the condition to use. For all parameters, click **Select values**. In all fields, but ORDERDATE, select the **Autoprompt** wizard. In the Autoprompt pane, choose **Select multiple values**. Then click **OK**.

Table 11-1 Parameters for WHERE conditions

	Field name	Condition	Select values
WHERE	PRODUCTTYPE	EQUAL to	▶ Select parameter ▶ Autoprompt ▶ Select multiple values
WHERE	PRODUCTCATEGORY	EQUAL to	▶ Select parameter ▶ Autoprompt ▶ Select multiple values
WHERE	SALESREP	EQUAL to	▶ Select parameter ▶ Autoprompt ▶ Select multiple values
WHERE	COUNTRY	EQUAL to	▶ Select parameter ▶ Autoprompt ▶ Select multiple values
WHERE	REGION	EQUAL to	▶ Select parameter ▶ Autoprompt ▶ Select multiple values
WHERE	STATE	EQUAL to	▶ Select parameter ▶ Autoprompt ▶ Select multiple values
WHERE	PLANTNAME	EQUAL to	▶ Select parameter ▶ Autoprompt ▶ Select multiple values
WHERE	ORDERDATE	GREATER THAN or EQUAL to	▶ Select parameter ▶ Enter field name From_Date under Parameter (do <i>not</i> autoprompt) See Figure 11-24
WHERE	ORDERDATE	LESS THAN or EQUAL to	▶ Select parameter ▶ Enter field name To_Date under Parameter (do <i>not</i> autoprompt)

- For the date fields, instead of using the Autoprompt wizard, enter a parameter name that you want to use (Figure 11-24). We use From_Date for the first date parameter and To_Date for the second date parameter.

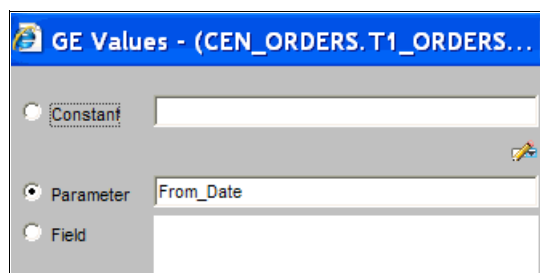


Figure 11-24 Date parameter

The finished Selection criteria pane should look like the example in Figure 11-25.

WHERE	(((	PRODUCTTYPE	EQUAL to	&PRODUCTTYPE PRODUCTTYPE IN CEN_ORDERS).P
WHERE	(((	PRODUCTCATEGORY	EQUAL to	&PRODUCTCATE PRODUCTCATEG CEN_ORDERS).P
WHERE	(((	SALESREP	EQUAL to	&SALESREP.(FIN CEN_ORDERS).S
WHERE	(((	COUNTRY	EQUAL to	&COUNTRY.(FIN CEN_ORDERS).C
WHERE	(((	REGION	EQUAL to	®ION.(FIND CEN_ORDERS).R
WHERE	(((	STATE	EQUAL to	&STATE.(FIND ST CEN_ORDERS).S
WHERE	(((	PLANTNAME	EQUAL to	&PLANTNAME.(F PLANTNAME IN CEN_ORDERS).P
WHERE	(((	ORDERDATE	GREATER THAN or EQUAL to	&From Date
WHERE	(((	ORDERDATE	LESS THAN or EQUAL to	&To Date

Figure 11-25 Selection criteria parameters

9. Save and close the report.

11.2.1 Creating the report in HTML Layout Painter

To create the report in HTML Layout Painter:

10. Navigate to the Tutorials folder in Developer Workbench. Right-click **Tutorials** and select **New** → **HTML File** (Figure 11-26).

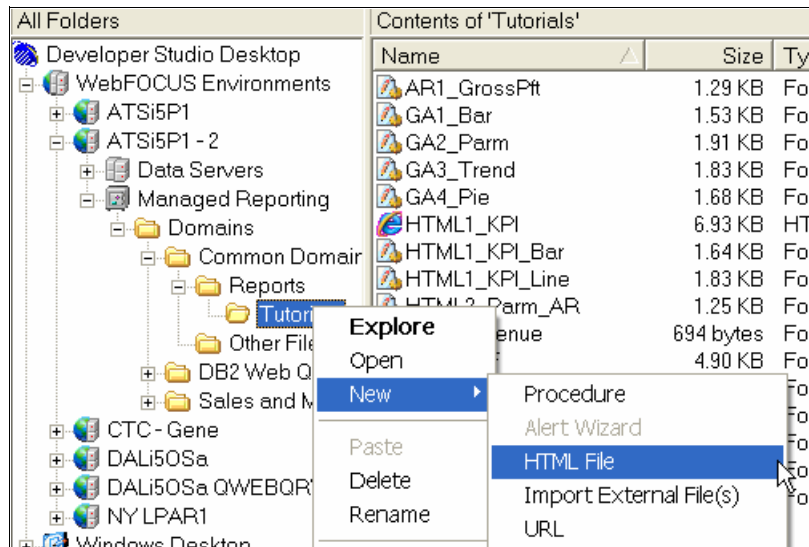


Figure 11-26 Creating a new HTML file

11. In the window that opens (Figure 11-27), enter a report name of HTML2_Parm and click **Create**.

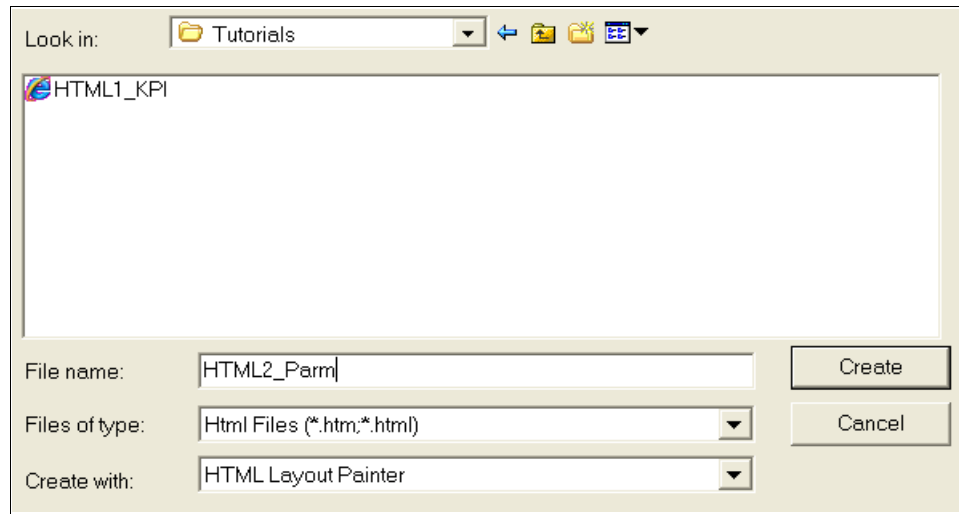


Figure 11-27 New report name

12. In this example, we create an HTML page that contains the various parameters and a Run button. When the Run button is clicked, a new window opens that contains the active report. We must import the report onto the HTML page so that DB2 Web Query automatically brings in all the parameters that are defined in the report, saving us from adding them manually. We remove the report frame itself after we import all the parameters.

Click the report icon (circled in Figure 11-28) and draw a small box in the upper left corner of the pane.

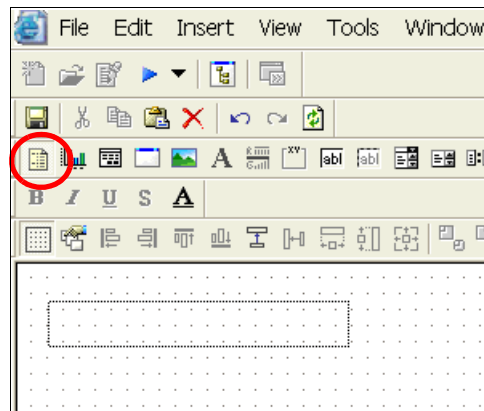


Figure 11-28 Importing a report

13. Right-click the report (the box that you just drew) and select **Reference existing procedure** (Figure 11-29).

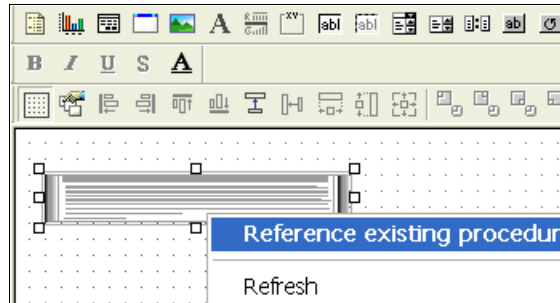


Figure 11-29 Link to existing procedure

14. Select the **HTML2_Parm_AR** report that you just created.

15. You are presented with a list of parameters that are referenced by this report (Figure 11-30).

- a. For SALESREP, change the control type to **Double list control**.
- b. For COUNTRY, change the control type to **Radio button**.
- c. For PLANTNAME, change the control type to **Check box**.
- d. For both dates, change the control type to **Calendar**.
- e. Under Parameter grouping options, select **Do not create a form**.

Parameters		
Name	Create control	Control Type
PRODUCTTYPE	☒	List box
PRODUCTCATEGORY	☒	List box
SALESREP	☒	Double list control
COUNTRY	☒	Radio button
REGION	☒	List box
STATE	☒	List box
PLANTNAME	☒	Check box
From_Date	☒	Calendar
To_Date	☒	Calendar

Parameter grouping options

Do not create a form

☐ Don't show this message again and use default selection

Figure 11-30 Selecting the parameter control types

The system lays out all the chosen parameters in a column on the HTML page similar to the layout in Figure 11-31.

The screenshot shows a web form with three sections, each containing a label and a list box. The background is a light gray grid. The first section is labeled 'Producttype:' and has a list box with 'ALL' selected, and 'Audio' and 'Camcorders' as other options. The second section is labeled 'Productcategory:' and has a list box with 'ALL' selected, and 'Amplifiers/PreAmps/Tuners' and 'Audio Systems' as other options. The third section is labeled 'Salesrep:' and has a list box with 'ALL' selected, and 'Adriano Prefumo' and 'Akinori Imai' as other options. To the right of the 'Salesrep:' list box, there are three buttons: '>', '<', and a button with 'Awwwww', 'Bwwwww', and 'Cwwwww' listed vertically.

Producttype:

ALL
Audio
Camcorders

Productcategory:

ALL
Amplifiers/PreAmps/Tuners
Audio Systems

Salesrep:

ALL
Adriano Prefumo
Akinori Imai

> < Awwwww Bwwwww Cwwwww

Figure 11-31 Imported parameters

16. Making sure that you keep the headings and controls together, drag the parameters into a layout similar to Figure 11-32. You must scroll to see the entire page of parameters. See the following steps to size the Country control box.

- Make sure that you leave enough room at the top of the report for a report heading.
- Some of the boxes, such as SALESREP, require resizing. Keep any frames that are within a panel, in the panel when they are moved or resized.

Figure 11-32 Rough layout for parameters

Note: This exercise is designed to introduce you to many of the controls and capabilities of HTML Layout Painter. Is it not intended to teach window design. Do *not* design a window that has this many different controls and looks like this in production applications.

17. We want to change Country from one column wide to two columns wide (Figure 11-33):

- a. Highlight the **Country** control box.
- b. From the Properties pane, select **columns** and type 2.
- c. Size the control box appropriately.

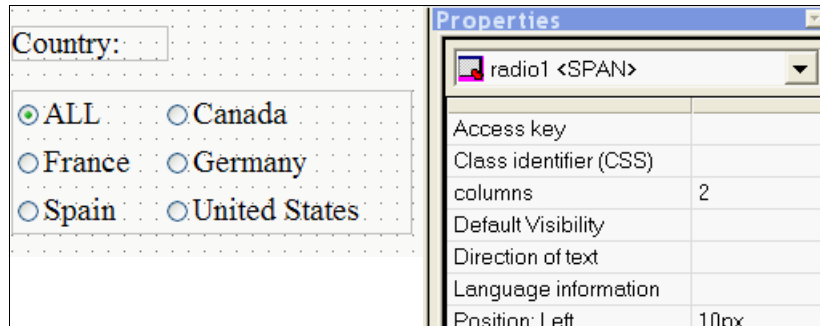


Figure 11-33 Two-column radio control box

Parameter controls

In this section, we illustrate how to define the sequence in which the cursor moves from cursor to cursor also known as *parameter controls*:

18. Highlight the **Producttype** control box.
19. In the Properties pane on the right, select **Tab index** and enter 1. Tab index controls the sequence in which the cursor moves from field to field.

20. Click the **Parameters** tab at the bottom of the right pane. This is where you can override options such as your type of control box or the text to go with the ALL option. Select the add **ALL** option for every parameter except the dates.
21. Highlight the **Productcategory** control box.
22. Click the **Properties** tab (Figure 11-34), and under Tab index, enter the value 2.

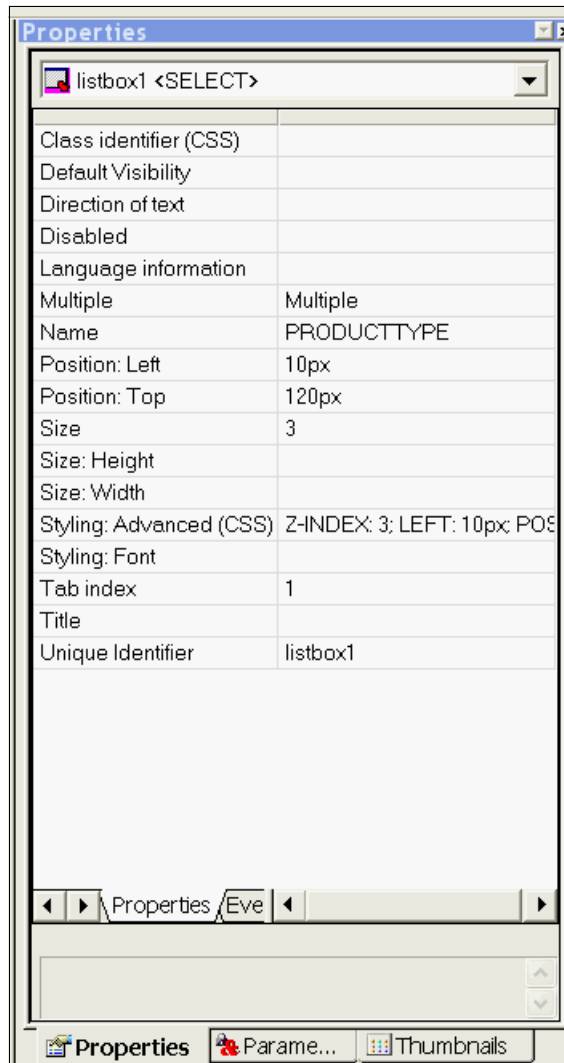


Figure 11-34 Properties tab

Chaining the controls together

We want to set up our parameters so that the value selected for product type modifies (filters) the values that are displayed in the product category drop-down list:

23. Click the **producttype** control box. Holding down the Shift key, also select **Productcategory**.

24. Click the **Add chain** button (circled in Figure 11-35). This function chains the parameters together in the sequence in which you selected them.

Tip: If you right-click a control box that is chained to another control, you can override the chaining logic. The default in our example shows all product categories that have a product type equal to what the user selected. You can choose any relationship to the controlling parameter, that is less than, greater than, and so on. In our example, EQUAL is really the only choice that makes sense and it is always the default.

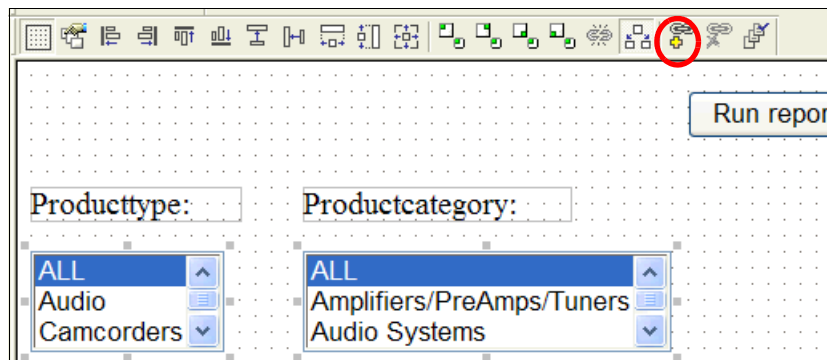


Figure 11-35 Chaining parameters

25. Highlight the **Salesrep** control. In the Properties pane, for Tab index, type 3.

26. Highlight the **Country** control. In the Properties pane, for Tab index, type 4.

27. Highlight the **Region** control. In the Properties pane, for Tab index, type 5.

28. Highlight the **State** control. In the Properties pane, for Tab index, type 6.

29. We want the Country parameter to control or filter the Region parameter, and we want the Region parameter to filter the State list box. To chain Country to Region to State:

- Select the **Country** control box.
- Hold down the Shift key, and select the **Region** control box. Still holding down the Shift key, select the **State** control box.
- Click the **Add chain** button shown in Figure 11-35.

30. Highlight the **Plantname** control. In the Properties pane, for Tab index, type 7.

Using calendar controls

In this section, we illustrate how to define the controls when dealing with Date fields:

31. Right-click the **calendar** icon under From_Date and select **Calendar Properties**.
32. For Date range, select **Static**. Specify a range from **January 1 2006** to **December 31 2007**. For Date format in data source, select **YYMD**. Click **OK**. See Figure 11-36.
33. Highlight the **From_Date** control.
34. In the Properties pane, for Tab index, type 8.
35. Repeat these calendar control steps for To_Date using a Tab index of 9.

Tip: The more common calendar option is probably relative dates, for example within the last two years. Since our data only contains dates in 2006 and 2007, we chose the static dates for this example.

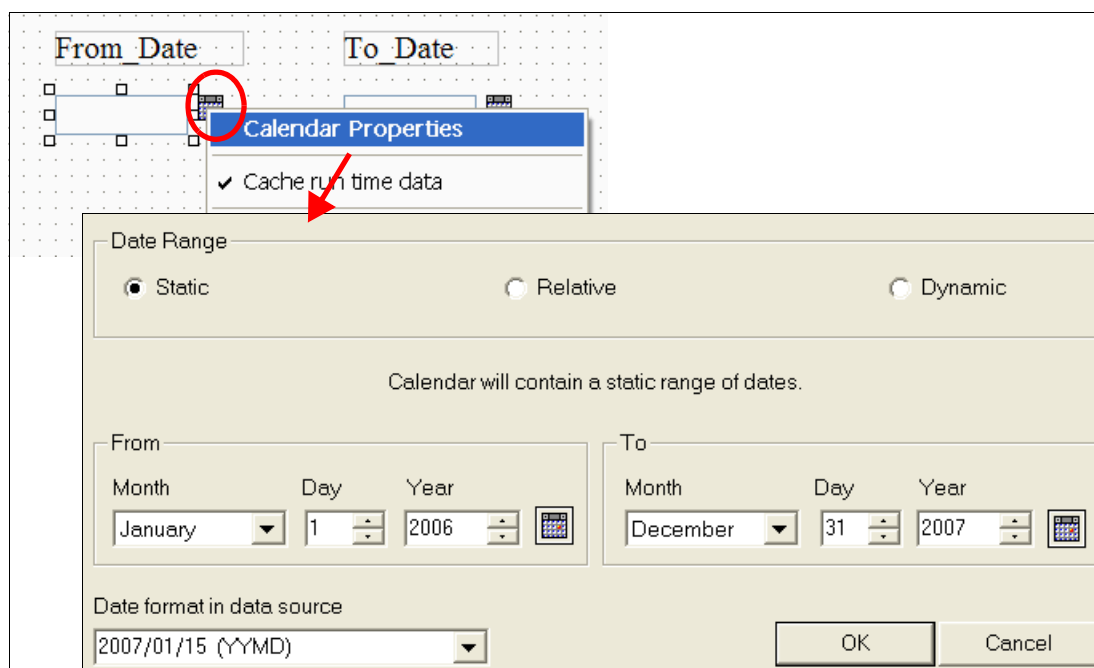


Figure 11-36 Calendar controls

36. Right-click the **Run report1** control box and select **Hyperlink Properties**.

37. In the Hyperlink Properties window (Figure 11-37), complete these steps:
 - a. Change Display Text to RUN.
 - b. Change Target Type to **Window**. This runs the report in a new window instead of in the frame drawn on the current HTML page.

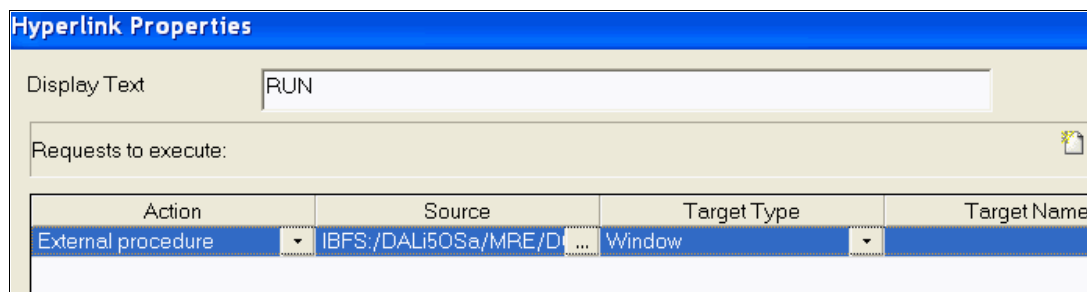


Figure 11-37 Run button hyperlink properties

38. Delete the original frame you drew when importing the report. It is located below all the parameters.
39. Select the **RUN** control box. In the Properties pane, for Tab index, type 10.
40. Resize the Run button by dragging the frame.
41. Add a heading to the report:
 - a. Click the **text** icon on the toolbar and draw a box to contain your heading (Figure 11-38).

Note: In addition to text and reports, you can import images to your page using the icon to the left of the text icon. You can use define cascading style sheets to tailor the look of your page.

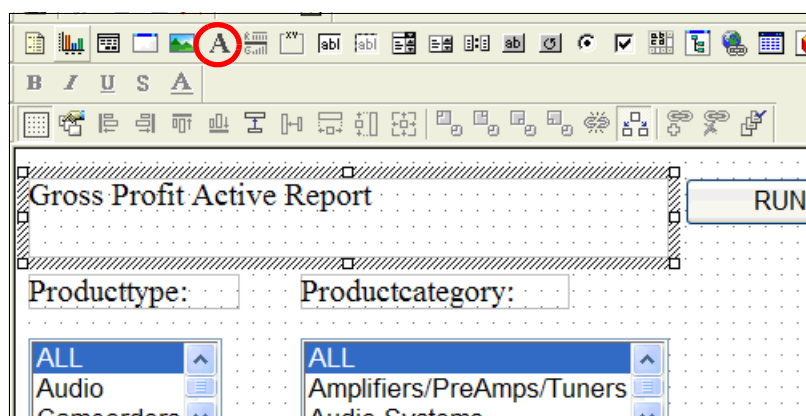


Figure 11-38 Adding text to a page

42. Enter a title of Gross Profit Active Report.
43. Click your **text box** so that it is surrounded by a straight line frame and not slashes.
44. In the Properties pane on the right, select the ... button next to Styling: Font.

45. In the Font window (Figure 11-39), select the styling for your heading:

- For Font, select **Arial**.
- For Font Style, select **Bold**.
- For Size, select **18pt**.
- For Effects, select **Underline**.
- Click **OK**.

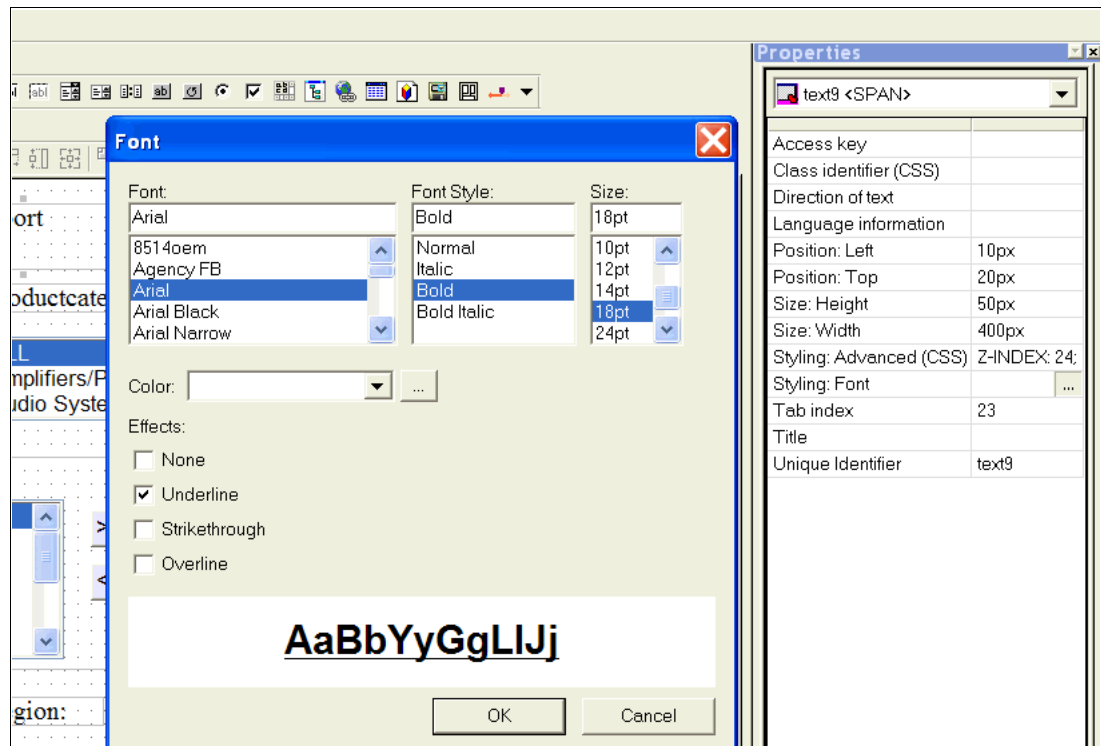


Figure 11-39 Styling text

46. Click the background area of the HTML layout.

47. In the Properties pane, click the ... button next to Background color and select a pale color (Figure 11-40).

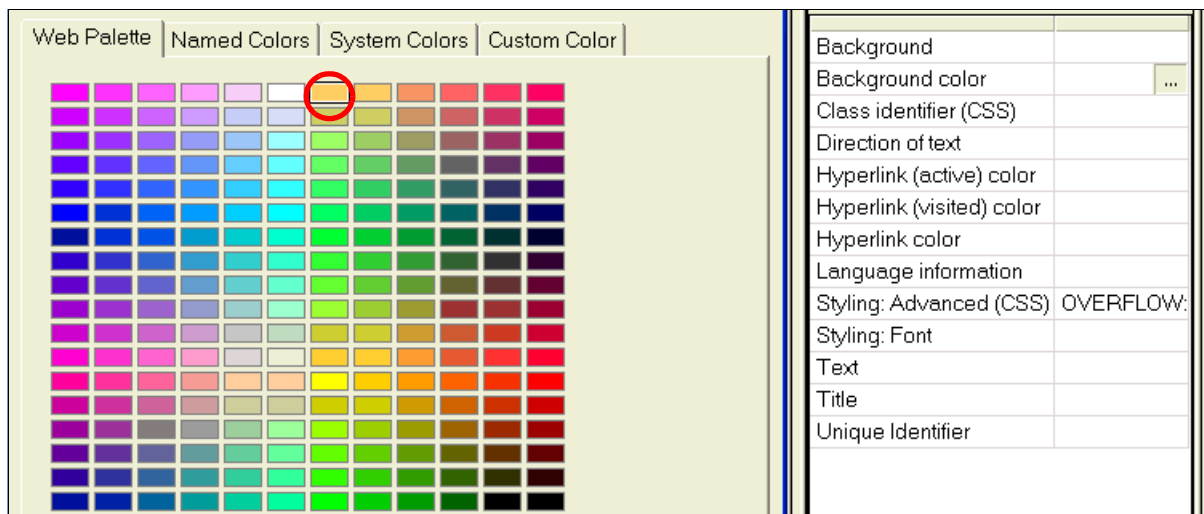


Figure 11-40 Background color

48. When we modified all the parameter control boxes, we changed the tab index in the Properties pane. This controls the sequence that the users tab through the fields in the window. We still have a problem in that some of the other elements on the page have a tab index that might conflict with our sequence numbers.

Starting with the title, highlight the frame, and find the tab index number. If it is less than or equal to 10, change it to an arbitrary number above 20. The Run button had a tab index of 10 and we want it to be last in our normal sequence. Do not forget to check the arrow elements in Salesrep.

Tip: If you hover over the icon second from the left that looks like a hand and multiple small boxes, you should see the text “Tab Order”. Selecting this shows the sequence of your tabs for the user. Make sure that you do not have duplicates and the order is left to right, top to bottom for INPUT capable fields.

49. Save and run the report.

50. For our first test, select **ALL** for Salesrep and **2006/01/01** to **2006/12/31** for the dates (Figure 11-41).

Tip: Having chosen the double list control for Salesrep means that you always have to select at least one item for SALESREP. This can be ALL, but there must be a value in the output list.

Gross Profit Active Report

RUN

Producttype:

ALL

Audio

Camcorders

Productcategory:

ALL

Amplifiers/PreAmps/Tuners

Audio Systems

Salesrep:

Adriano Prefumo

Akinori Imai

Alex McKay

Andreas Graff

Andrew Trousdale

Andy Carter

>

<

ALL

Country:

☒ ALL

☐ Canada

☐ France

☐ Germany

☐ Spain

☐ United States

Region:

ALL

Andalucia

Cataluna

State:

ALL

Alabama

Arizona

Plantname:

☒ ALL

☐ Boston

☐ Dallas

☐ Los Angeles

☐ Orlando

☐ Saint Louis

☐ Seattle

From_Date

2006/01/01

To_Date

2006/12/31

Figure 11-41 Prompting for parameters

51. Click **RUN**. Figure 11-42 shows the output of running this report.

Gross Profit for Type Audio				
Product Type ▼	Product Category ▼	Revenue ▼	Cost of Goods Sold ▼	Gross Profit ▼
Audio	Speakers	40,071,006.00	11,753,970.00	\$28,317,036
	Audio Systems	64,356,707.00	43,278,460.00	\$21,078,247
	CD Players and Recorders	27,740,934.00	19,325,220.00	\$8,415,714
	MP3	20,764,481.00	12,450,660.00	\$8,313,821
	Amplifiers/PreAmps/Tuners	19,896,918.00	12,100,720.00	\$7,796,198
	Receivers	18,090,638.00	11,673,250.00	\$6,417,388
Camcorders	DVD Camcorders	185,323,541.00	146,565,450.00	\$38,758,091
	MiniDV Camcorders	24,961,102.00	16,557,150.00	\$8,403,952
	Digital8 Camcorders	6,232,363.00	2,987,110.00	\$3,245,253
Cameras	Digital Cameras	90,761,413.00	65,764,150.00	\$24,997,263
Office	Organizers	5,648,224.00	2,385,290.00	\$3,262,934
	Handheld and PDA	9,010,664.00	6,844,330.00	\$2,166,334
Video	DVD	160,057,256.00	120,685,650.00	\$39,371,606
	TV	82,297,472.00	73,378,600.00	\$8,918,872
	VCR	11,089,989.00	8,284,340.00	\$2,805,649

Figure 11-42 Final output

At this point, you can try and run the report taking advantage of the parameter chaining. Selecting a product type changes the product categories list box. Similarly choosing a region changes the state displayed.

Then go back to your parameter design page and start tailoring the page to look like a more practical report that you might show to a user.



Additional features of Developer Workbench

In this chapter, we cover the following additional features of Developer Workbench:

- ▶ Synonym Editor
 - Metadata
 - Date decomposition
 - Creating filters
 - Joining tables
 - Business views
 - Impact analysis
 - Data profiling
- ▶ SQL Wizard

We provide a short tutorial for each feature so you can see how to use it. There are many more capabilities within each of the features than is described in this chapter. See the online help text in Developer Workbench for more details.

12.1 Synonym Editor

DB2 Web Query can create basic metadata from the DB2 Web Query home page or via the Synonym Editor in the optional, but highly recommended, Developer Workbench tool. When the basic metadata exists, you must use the Synonym Editor if you want to enhance the metadata or take advantage of the more advanced metadata capabilities. Within Synonym Editor, you can change the way a field is displayed to the users. In our previous examples, if we add commas and floating decimal signs to LINETOTAL in the metadata, then the developers never need to do this in the individual reports. Other examples include converting date formats, converting alpha to numeric fields, or changing the length of a field.

The Synonym Editor also allows you to create new calculated fields for the users. This includes the ability to break dates down into their various components. If you plan to use the optional online analytical processing (OLAP) feature, you must define your hierarchies or dimensions through the Synonym Editor. For more details about defining hierarchies for OLAP use, see Chapter 10, “Online analytical processing” on page 201.

Tip: Basic synonyms can be created from within Developer Workbench using the same routines that are used from the DB2 Web Query home page. Right-click **baseapp** and select **New** → **Synonym** to invoke the metadata creation windows.

1. To start the Synonym Editor, expand the metadata portion of the WebFOCUS Environments. Expand **WebFOCUS Environments** → **your environment** → **Data Servers** → **EDASERVE** → **Application** → **baseapp**.
2. In the right pane, select the table that you want to open in Synonym Editor. In this tutorial, we select **cen_orders.mas**. Right-click and select **Edit in Synonym Editor** (Figure 12-1).

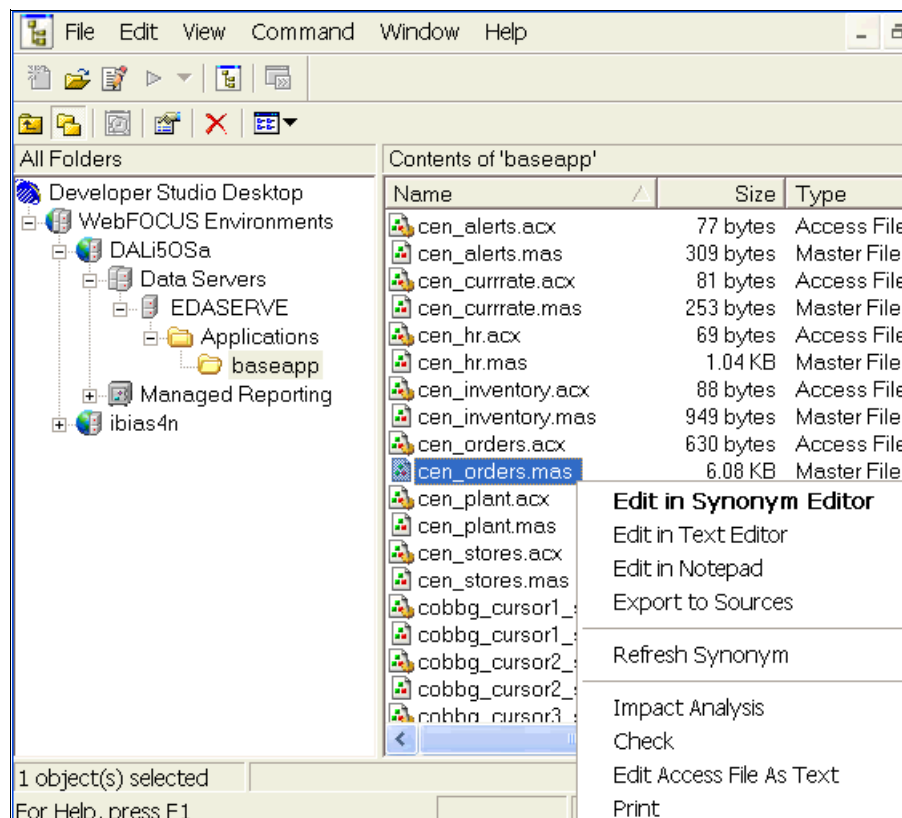


Figure 12-1 Starting the Synonym Editor

The window changes to show the different attributes of the table that are stored in the metadata (Figure 12-2). Look at the file attributes that are available. Some of the attributes, such as DBSPACE, are not applicable to DB2 for i5/OS.

Important: The Synonym Editor does not allow you to make any changes that render the Master file unusable.

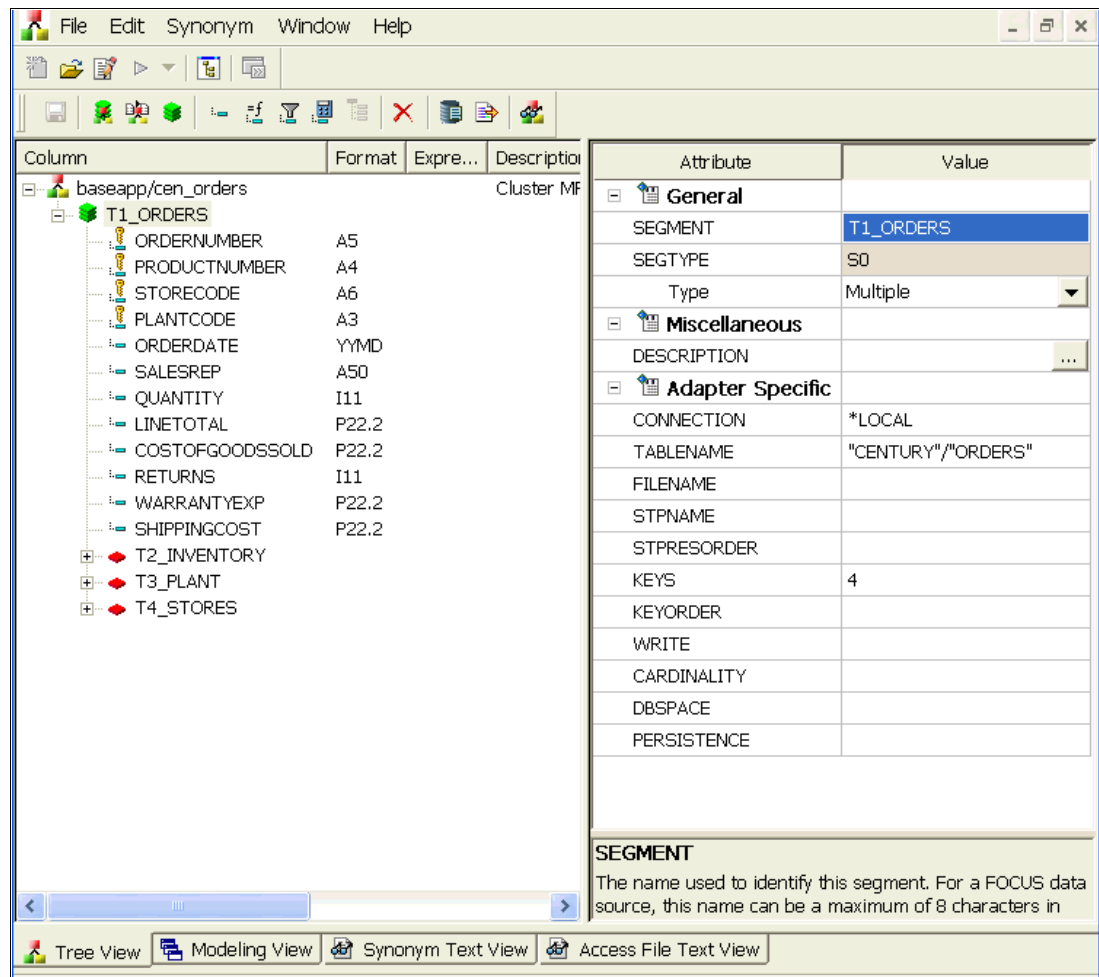


Figure 12-2 Metadata: Orders master file

- Highlight **LINETOTAL** to see the metadata attributes of the field displayed (Figure 12-3). In this example, we want LINETOTAL to display in reports with commas and a floating dollar sign. When we make this change, the report will use these settings by default.

Tip: The display in Figure 12-3 also allows users to specify a description and title for a column. User's can set up multiple titles for a column based on the language of the user. Selecting the ellipsis (three dots) next to these attributes allows you to specify which title you want displayed based on the language chosen.

Column	Attribute	Value
baseapp/cen_orders T1_ORDERS ORDERNUMBER PRODUCTNUMBER STORECODE PLANTCODE ORDERDATE SALESREP QUANTITY LINETOTAL COSTOFGOODSSOL RETURNS WARRANTYEXP SHIPPINGCOST ORDERDATE_DAY ORDERDATE_MONT ORDERDATE_YEAR ORDERDATE_QUAR T2_INVENTORY T3_PLANT T4_STORES	General	
	FIELDNAME	LINETOTAL
	ALIAS	LineTotal
	MISSING	☒
	TITLE	Line Total
	ACTUAL	P11
	Type	Decimal Packed
	Length	11
	Decimal	0
	USAGE	P22.2CID
	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	!D - Floating dollar sign
	Currency Symbol	!I - Fixed pound sterling sign
	Extended Currency Symbol Display Op	!L - Floating pound sterling sign
		!y - Fixed Japanese yen symbol
		!Y - Floating Japanese yen symbol
		!d - Fixed dollar sign
		!D - Floating dollar sign

Figure 12-3 Field- and column-level attributes

Another interesting attribute on this display is the USAGE attribute (Figure 12-4). To demonstrate this, highlight and select **ORDERDATE** in the Column pane.

Attribute	Value
 General	
FIELDNAME	ORDERDATE
ALIAS	OrderDate
MISSING	☒
TITLE	Order Date 
ACTUAL	DATE
Type	Date 
Date	
USAGE	YYMD
Type	Date 
Date Components	Three-part: YYMD, MDYY, DMY
Date Order	YYMD - 2004/08/21 
Year	YY - 4 digit year - 2004 
Month	M - 2 digit month 
Day of the Week	
Separator	/ - 2004/08/21 
DEFCENT	0
YRTHRESH	0

Figure 12-4 Date field attributes

In the Attribute pane, ACTUAL refers to how a column is stored in DB2 for i5/OS. USAGE refers to how the user sees and works with a column. USAGE allows us to override or change the length and attributes of a column from how it is stored and displayed by default.

For many System i customers, USAGE becomes important for date fields. If you have a field defined to DB2 as an integer 8 field that happens to contain the value 20070901, there is no automatic way for DB2 Web Query to know that this is a date field. The ACTUAL for this column displays as integer 8. Using the powerful USAGE attribute allows you to specify a USAGE of type Date with a format of YYMD. This enables many of the date and time functions, calculations, and formatting options that are available in DB2 Web Query to work with your integer field.

12.1.1 Date Decomposition

In many of the reports we have created so far, we have broken the date field into years, quarters, and months. This is a common requirement. To save you from doing this manually in every report, DB2 Web Query provides the Date Decomposition feature to automatically decompose the date in the Synonym Editor.

4. Right-click the **ORDERDATE** field and select **Decompose Date** (Figure 12-5).

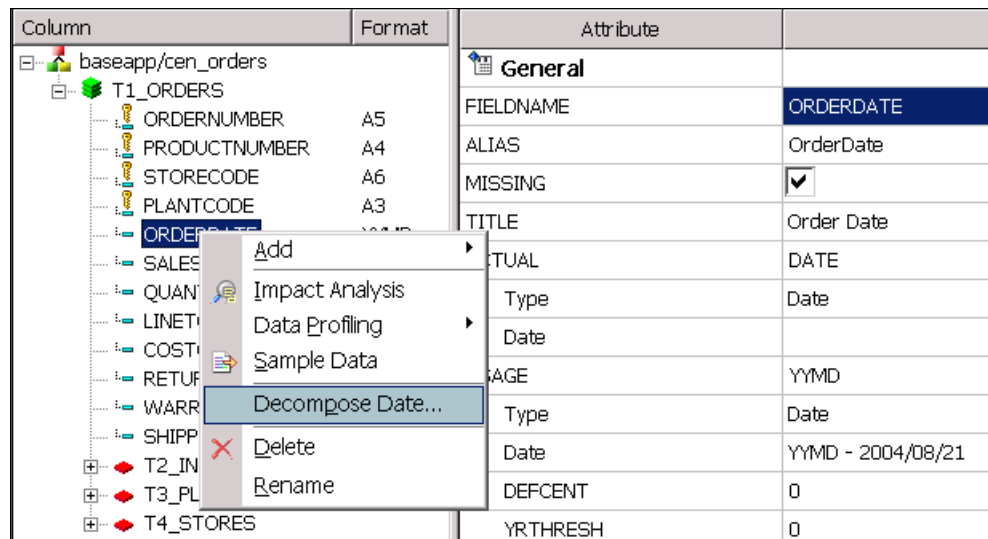


Figure 12-5 Decompose date column

5. In the Decompose Date Column window (Figure 12-6), select all four subcomponents of ORDERDATE to be created for you. Click **OK**.

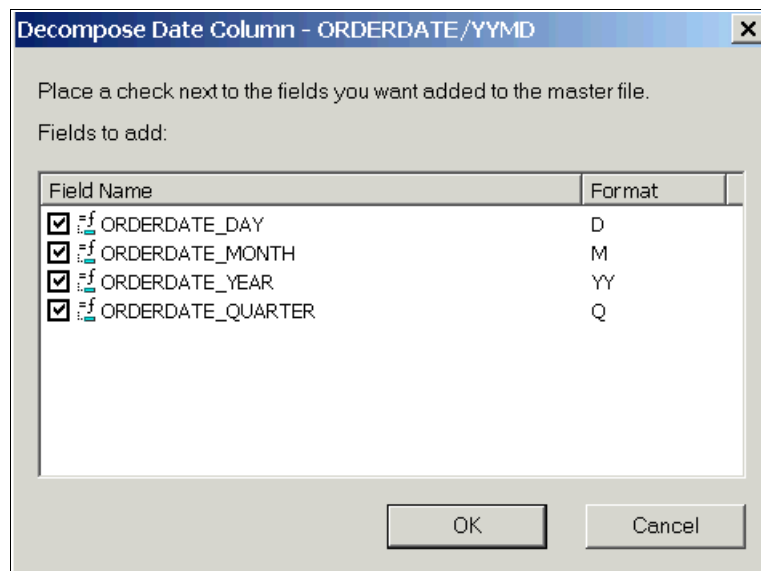


Figure 12-6 Decomposed date

In the results shown in Figure 12-7, notice that the system has generated four new fields for the users to access.

Column	Format	Expression
baseapp/cen_orders		
T1_ORDERS		
ORDERNUMBER	A5	
PRODUCTNUMBER	A4	
STORECODE	A6	
PLANTCODE	A3	
ORDERDATE	YYMD	
SALESREP	A50	
QUANTITY	I11	
LINETOTAL	P22.2C!D	
COSTOFGOODSSOLD	P22.2	
RETURNS	I11	
WARRANTYEXP	P22.2	
SHIPPINGCOST	P22.2	
ORDERDATE_DAY	D	ORDERDATE
ORDERDATE_MONTH	M	ORDERDATE
ORDERDATE_YEAR	YY	ORDERDATE
ORDERDATE_QUARTER	Q	ORDERDATE
T2_INVENTORY		
T3_PLANT		
T4_STORES		

Figure 12-7 Results of decomposing a date

Note: You can manually create these fields by right-clicking a column or the table and specifying **Add** → **Virtual column**. This allows you to create any new column that you normally create during the report definition stage. Creating it here means that it is available to all reports.

Creating a column, such as PROFIT, in the metadata means that the definition of the column is standard across all your reports.

12.1.2 Additional date manipulation

Our QWQCENT database stores dates in the DATE format. We know that many System i customers do not store dates in the DATE format. We add a few suggestions for how you might want to work with dates that are stored in alphameric or numeric formats. For example, 20070914 might be stored as a char 8 field.

First you highlight your date column. In this case, the date field is called DATEA8. Then modify the ACTUAL attribute from A8 to A8YYMD assuming that your date is stored in the format 20070914. You can also modify it to A8DMYY (Figure 12-8 on page 272) if you stored your date as 14092007. If you have a six-character field, you change the YY to simply Y. That is, you tell the system that you have a two-character year instead of a four-character year.

This also works if your date is stored as a numeric or packed field.

Attribute	
 General	
FIELDNAME	DATEA8
ALIAS	DATEA8
MISSING	☒
TITLE	
ACTUAL	A8YYMD
Type	Date
Date	
USAGE	A8YYMD
Type	Date
Date Components	Three-part: YYMD, MDYY, DMY
Date Order	DMYY - 21/08/2004

Figure 12-8 Defining your current character or numeric date field

Having made these changes, your users can now do basic date formatting. In a report, the user can choose to format the field as a YYMTRD field. This prints the date in the report as 2007, SEPTEMBER 14. The user can also create separate year, month, and day fields from this date.

To use various date functions and date arithmetic, you create a true DATE field. That is, you create a define field with your new date field name; the format is similar to YYMD. The expression is your original numeric or alpha date field that you formatted as A8YYMD.

As shown in Figure 12-9, you can now use the PROPERDATE field to do date arithmetic or use date functions such as adding seven business days to a date or finding the end of the next quarter. These functions are described in “Date functions” on page 371.

Column:	PROPERDATE	Format:	YYMD	...
Expression Relational Expression				
DATEA8				

Figure 12-9 Redefining a character date

If you use two-digit years, you can use the DEFCENT and YRTHRESH column attributes to define your century. If the year is greater than or equal to the threshold (YRTHRESH), then the two additional significant digits in the year become equal to the default century value (DEFCENT). If your two-digit year is less than YRTHRESH, then the additional two digits equal DEFCENT plus one.

Assume YRTHRESH equals 40 and DEFCENT equals 19. Any year from 40 to 99 is translated to 1940 - 1999. Any year from 00 to 39 is assumed to be 2000 - 2039. If you use a century field, you can create your own formula in a define field to calculate a new date field correctly.

If you store year, month, and day in individual fields, you must first create an interim field that has the format of A8YYMD (or A6YMD) and equals YRFLD | MTFLD | DAYFLD. Then, using this interim field, you create a new date field as shown in Figure 12-9.

After you create your new date field, this is the one that the users use in all their reports. This format opens up many additional date formatting and arithmetic functions.

12.1.3 Creating filters

You can predefine standard selection criteria in the master file. This allows the criteria to be specified once and used in multiple reports. In this example, we create a filter called Europe. This allows us to easily run a report including European only countries.

6. Right-click the **master file name** or a **field name** and select **Add** → **Filter** (Figure 12-10).

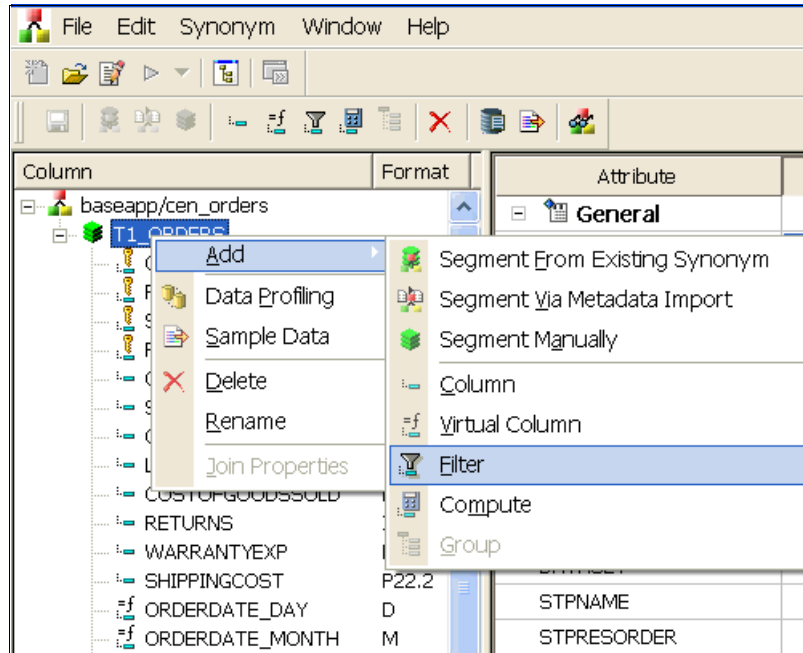


Figure 12-10 Adding a filter to a master file

7. In the Filter Calculator window (Figure 12-11), for Column, type Europe, and define it as an I1 (integer one long) column. This means that we can test if for true or false in our reports.

Tip: Clicking the plus sign (+) next to a field name to expand it returns a list of distinct values in that field. You can then drag this list of values to your filter expression.

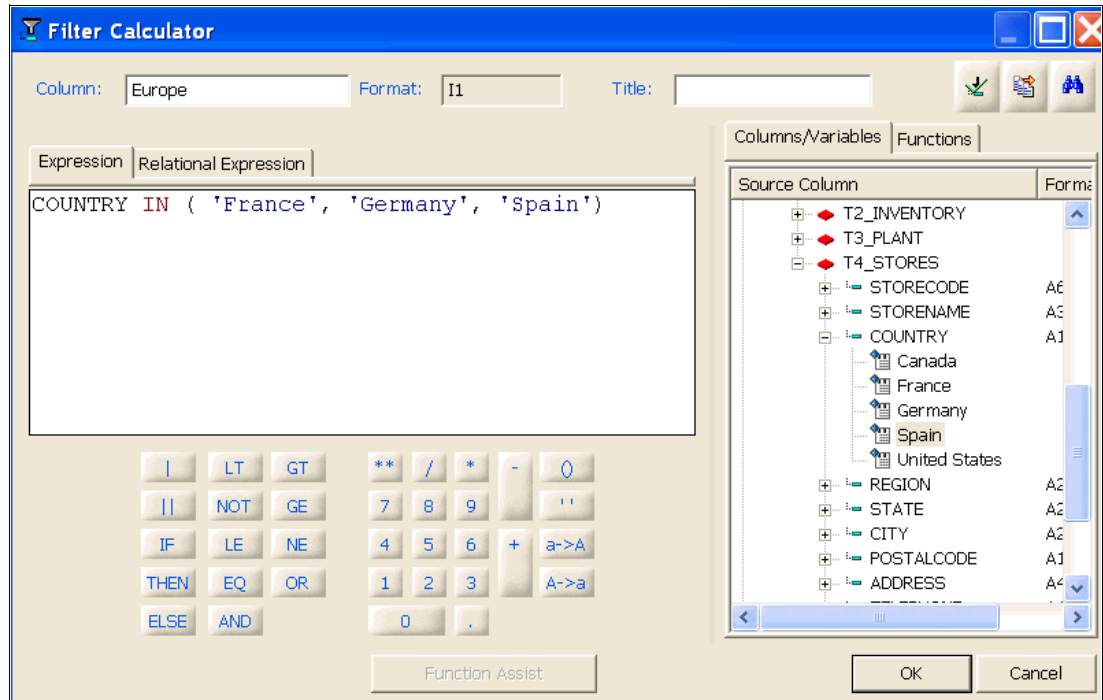


Figure 12-11 Creating a filter

8. The filter is displayed in your list of column names when you create a report. Drag the filter to the Selection criteria pane. A “WHERE filter is true” statement is generated by default. This report now includes only the countries that are defined in the Europe filter (Figure 12-12).

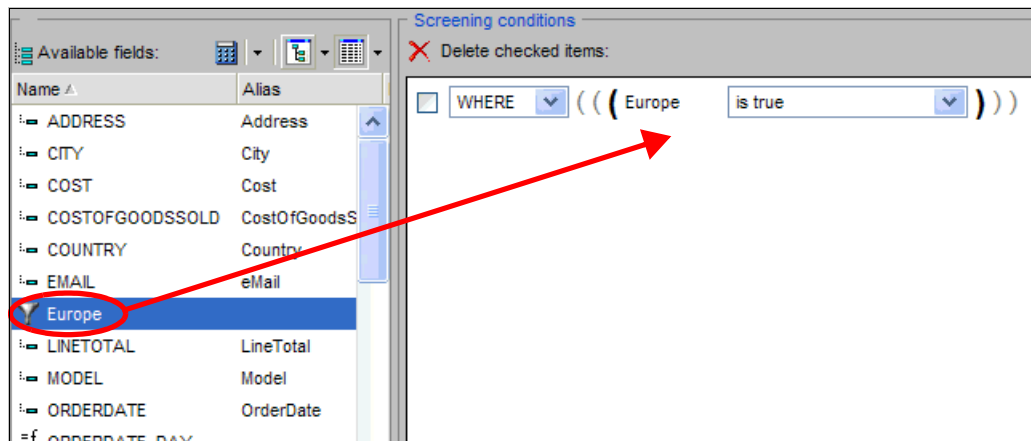


Figure 12-12 Using a predefined filter

12.1.4 Joining tables

If you are unable to define foreign keys in your DB2 tables, you can accomplish the same result for users who are using Developer Workbench. In this example, we start by working with the Orders table without using foreign key support. We then join this table to the inventory table that we already defined. This create a *cluster join* where the two tables look like one to the user.

1. Create another synonym for the Orders table, but without selecting Foreign Key support. Use either the DB2 Web Query home page or from within Developer Workbench.

In Developer Workbench, expand **WebFOCUS Environments** → **your environment** → **Data Servers** → **EDASERVE** → **Applications** → **baseapp**. Right-click and select **New** → **Synonym** (Figure 12-13).

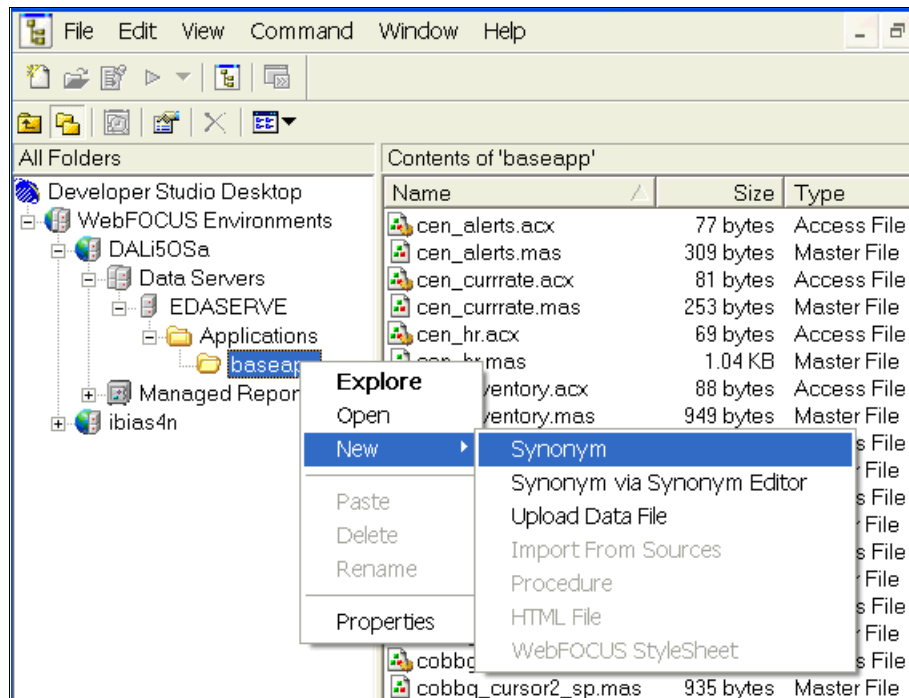


Figure 12-13 Creating synonyms from Developer Workbench

2. In the Create Synonym window (Figure 12-14), since our table is a normal DB2 for i5/OS table stored on our local system, select ***LOCAL** and click **OK**.

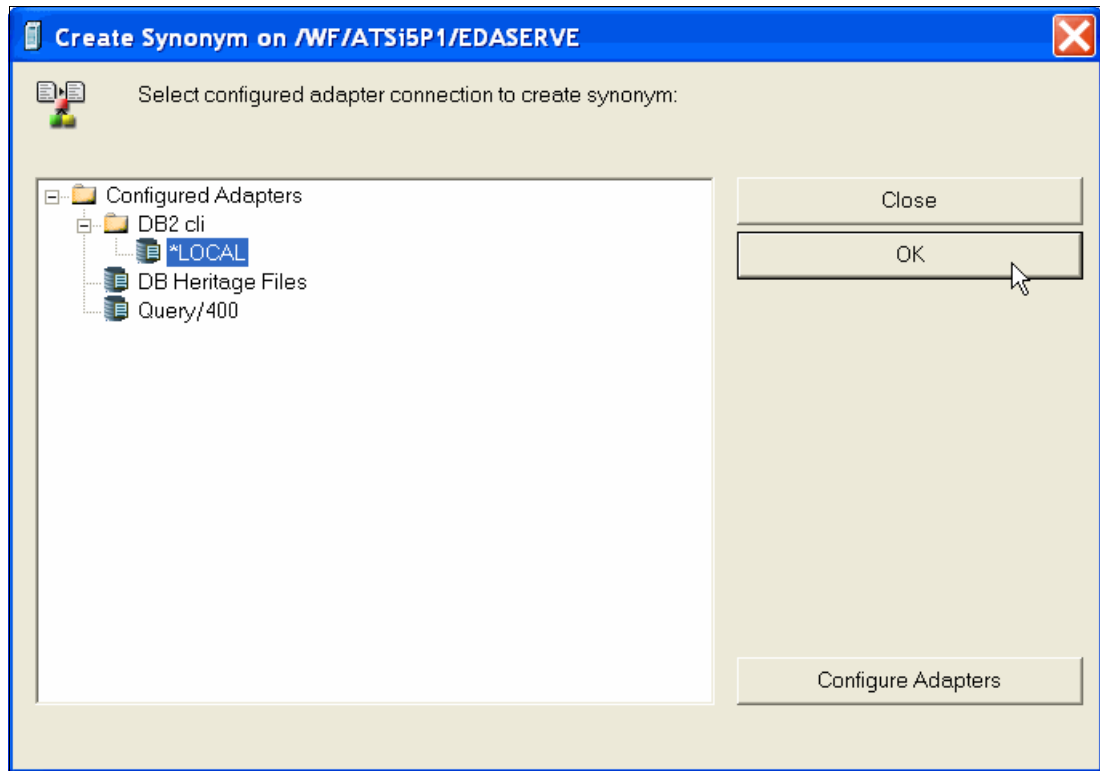


Figure 12-14 Creating metadata for local DB2 tables

3. In the next window (Figure 12-15), specify that the candidates for synonyms are the list of tables in the QWQCENT library. Select the **Tables** check box and make sure that the Views check box is cleared. For Library, select **qwqcent**. Then click **Next**.

DB2 Web Query 763 Server on ATSi5P1.RCHLAND.IBM.COM

Select Synonym Candidates for DB2 cli (*LOCAL) Step 1

Restrict object type to	
Restrict object type to	Tables, Views and Other Objects
Further restrict search to:	
Tables	☒
Views	☐
Aliases	☐
MQTs	☐
Library	qwqcent
Object name	

Next>> Cancel

Figure 12-15 Filtering synonym candidates

- In the next window (Figure 12-16), for Select application, ensure that **baseapp** is selected and select only the **ORDERS** table. Do not select “with foreign keys”.

We do not need to create a synonym for the INVENTORY table as we did that in an earlier exercise.

Click the **Create Synonym** button.

DB2 Web Query 763 Server on ATSi5P1.RCHLAND.IBM.COM

Create Synonym for DB2 cli (*LOCAL) Step 2 of 2

Select options on the left then indicate in the table below for which items to Create Synonym using these options.

Cardinality ☐ With foreign keys ☐ Dynamic columns ☐ One-part name ☐ Select application **baseapp** Prefix Suffix Customize data type mappings ☐ Overwrite existing synonyms ☐

☐	Default synonym name	Library/Schema	Table name	Type
☐	ALERTS	QWQCENT	ALERTS	TABLE
☐	CURRRATE	QWQCENT	CURRRATE	TABLE
☐	HR	QWQCENT	HR	TABLE
☐	INVENTORY	QWQCENT	INVENTORY	TABLE
☒	ORDERS	QWQCENT	ORDERS	TABLE
☐	PLANT	QWQCENT	PLANT	TABLE
☐	QRPGLSRC	QWQCENT	QRPGLSRC	TABLE
☐	QWQCENT	QWQCENT	QWQCENT	TABLE
☐	STORES	QWQCENT	STORES	TABLE

Back Create synonym Cancel

Figure 12-16 Selecting tables

5. After you see the window that indicates that the synonym for the ORDERS table was created successfully, click **Open** (Figure 12-17).

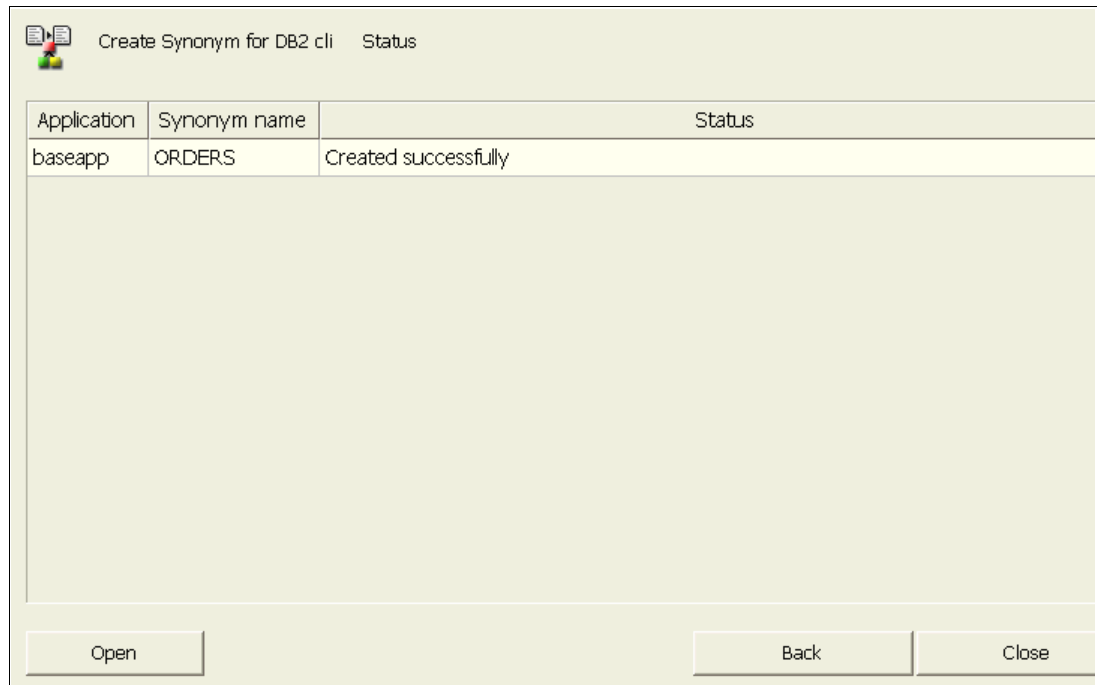


Figure 12-17 Successful creation of a synonym

6. In Developer Workbench, right-click **ORDERS** and select **Add → Segment from Existing Synonym** (Figure 12-18).

Note: You can add your join from the Modeling view tab at the bottom of the window as well as from the Tree View. Most of the steps are identical. From the Modeling view, right-click the **table** icon and select **Add → Segment from Existing Synonym**.

If metadata for INVENTORY is not yet created, you can select **Add → Segment Via Metadata Import**.

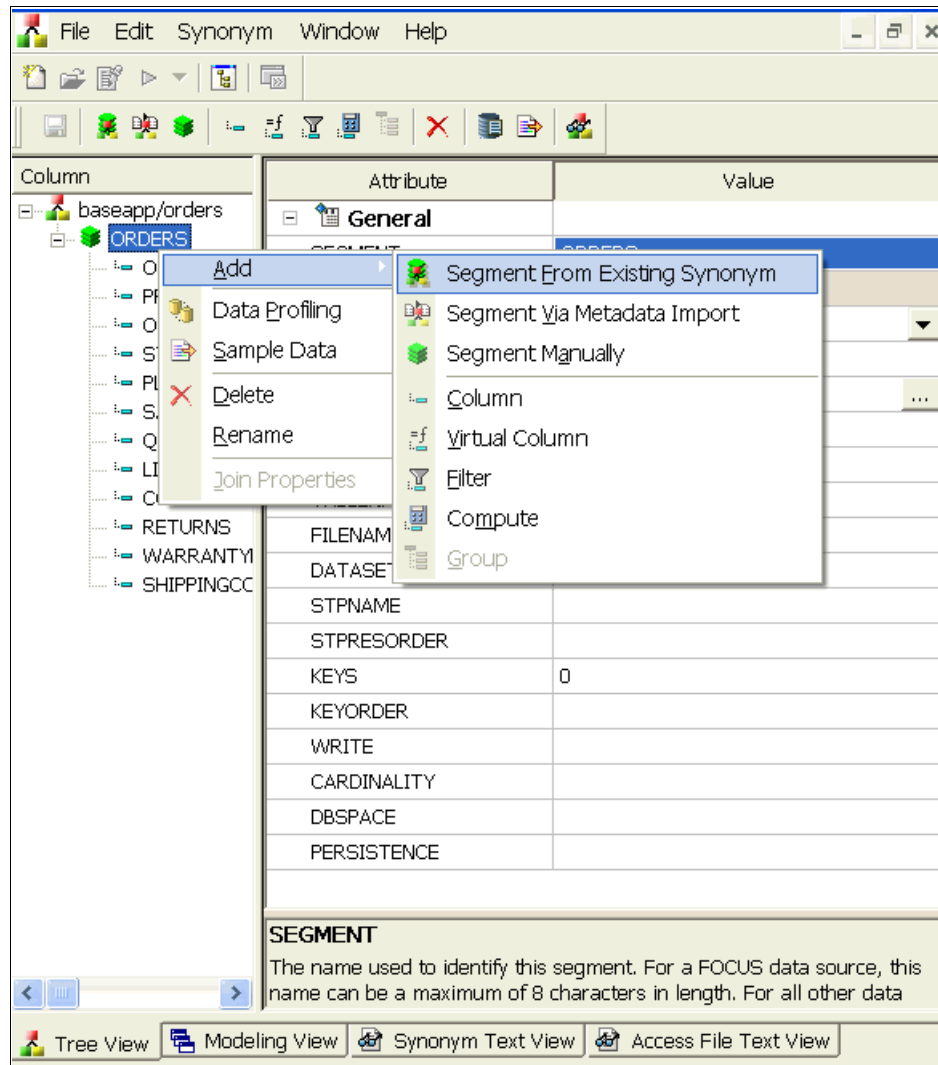


Figure 12-18 Adding a segment or secondary table

7. In the Select Synonym window (Figure 12-19), select **cen_inventory** as the table to join to ORDERS. Click **Select**.

DB2 Web Query creates a default join between the two tables.

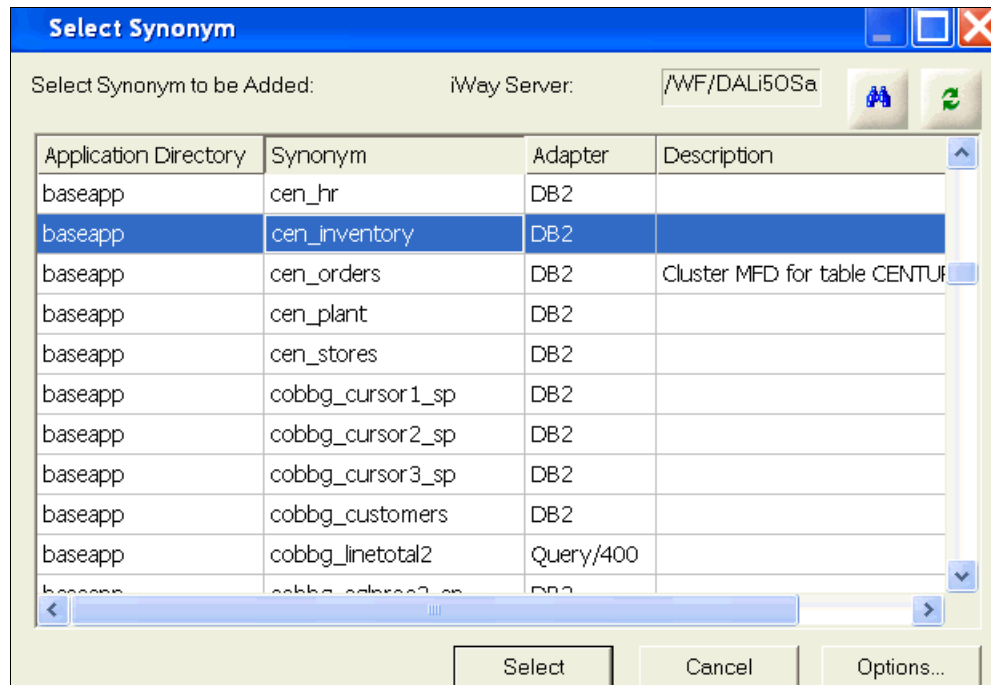


Figure 12-19 Selecting the secondary join table

8. In the Developer Workbench window (Figure 12-20), in the right pane, for SEGTYPE, the join defaults to Multiple or a one-to-many join. If you want to specify a one-to-one join, change Multiple to **Unique**.

In the left pane, to see and possibly modify the join type and join conditions, right-click **CEN_INVENTORY** and select **Join Properties**.

Tip: If you are using the Modeling view, you can either click the file icon or you can double-click the arrowed lines between the files to see the Join Properties.

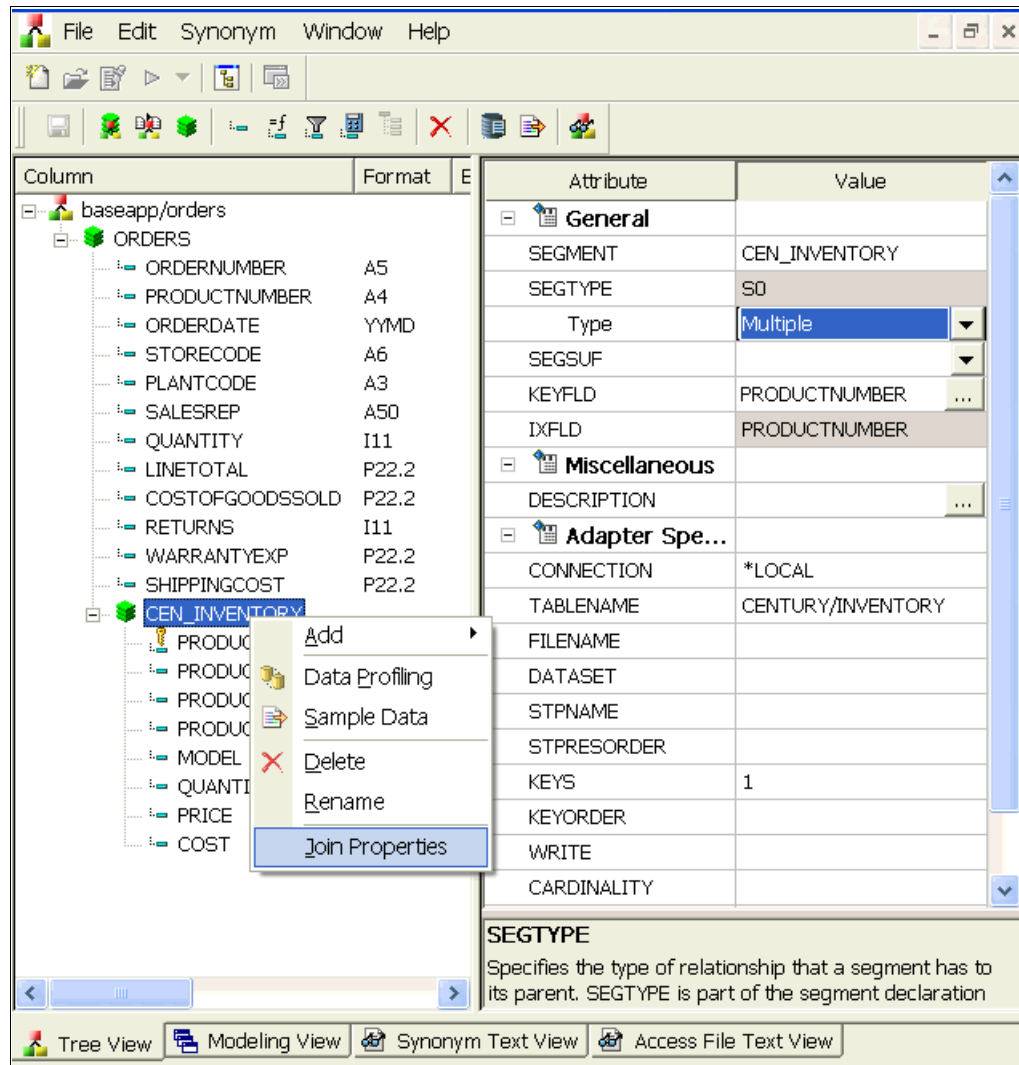


Figure 12-20 Specifying Join Properties

- In the Join Properties window (Figure 12-21), notice that DB2 Web Query has chosen an Inner Join, where Product Number from both tables is equal.

From this window, you can override the Join Type and Join Conditions you require. Click **OK**.

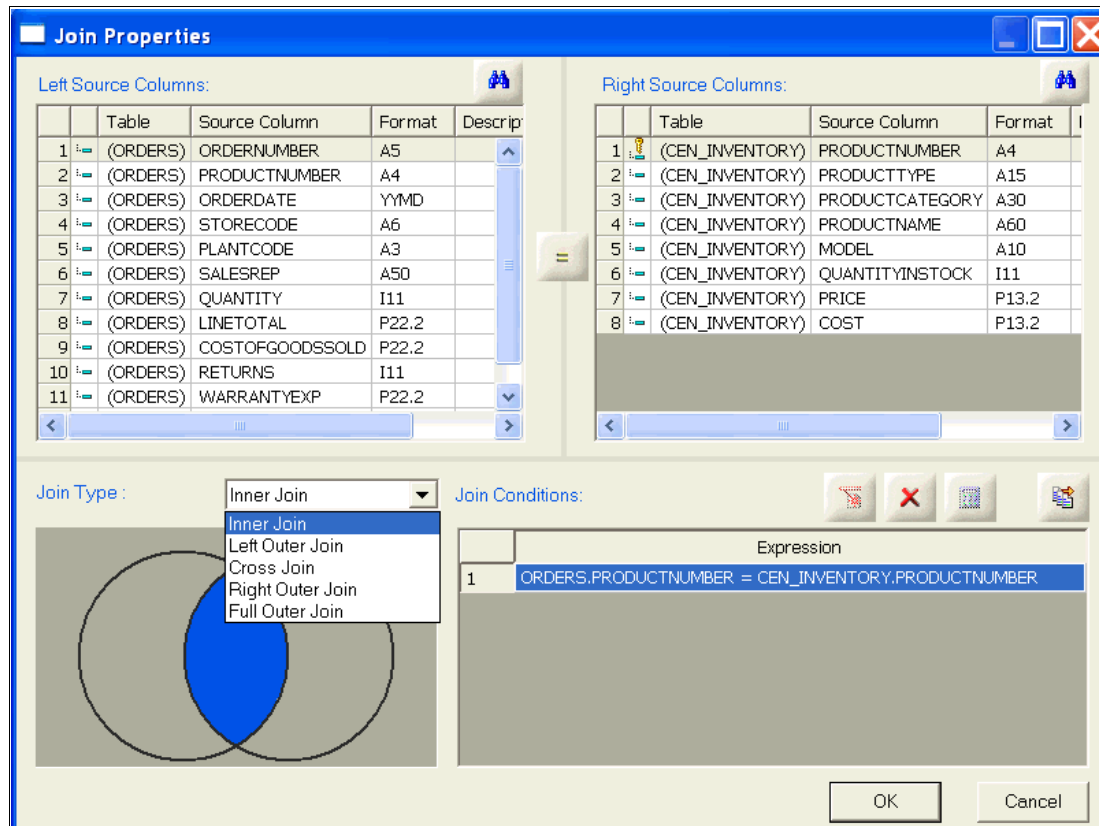


Figure 12-21 Default join

12.1.5 Business View

The Business View allow you to define a logical view of your data typically to simplify ad hoc reporting for the user. With the Business View, you can simplify the fields in your master file.

In a view, fields can be grouped into folders for logical partitioning of the data elements. You can choose to have a folder that contains all the elements of an employee's address or you might want a folder that stores all your filters selected from the master file synonym.

The Business View also allows you to specify changes to your underlying table, such as new column headings, format changes, and so on. These changes are not lost if you need to reimport the underlying table. If you make these changes in the master file for the base or underlying table and you must recreate the synonym for that table, you lose many of the attributes that you changed.

In addition, the Business View enables you to add define and compute fields to a table so that all users have access to them in their reports. The new define fields or virtual columns can contain system variables such as &DATE or the current user name. The current user name can be retrieved using the GETUSER function.

1. From the Synonym Editor (Figure 12-22), right-click **cen_orders** and select **Create Business View**.

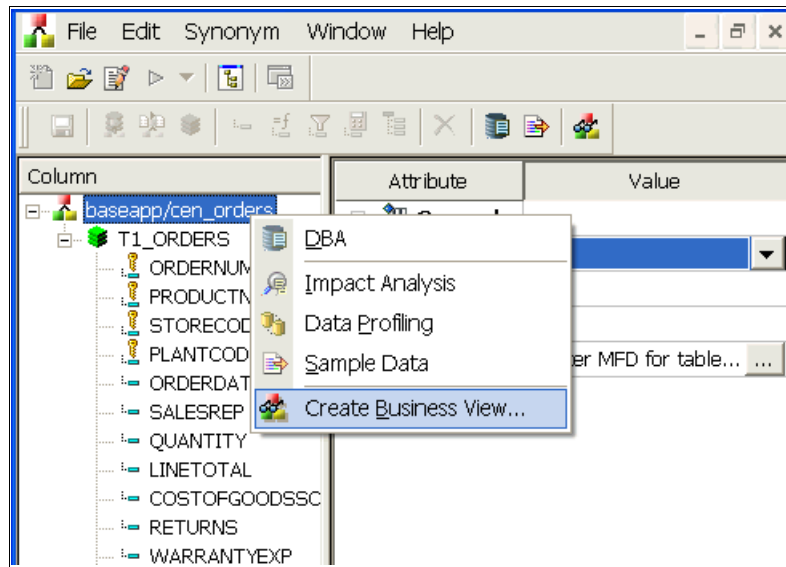


Figure 12-22 Creating a Business View

2. In the next window, you can create folders (right-click) and drag columns, defined fields, computed fields, or filters to your view.

Create the business view shown in Figure 12-23.

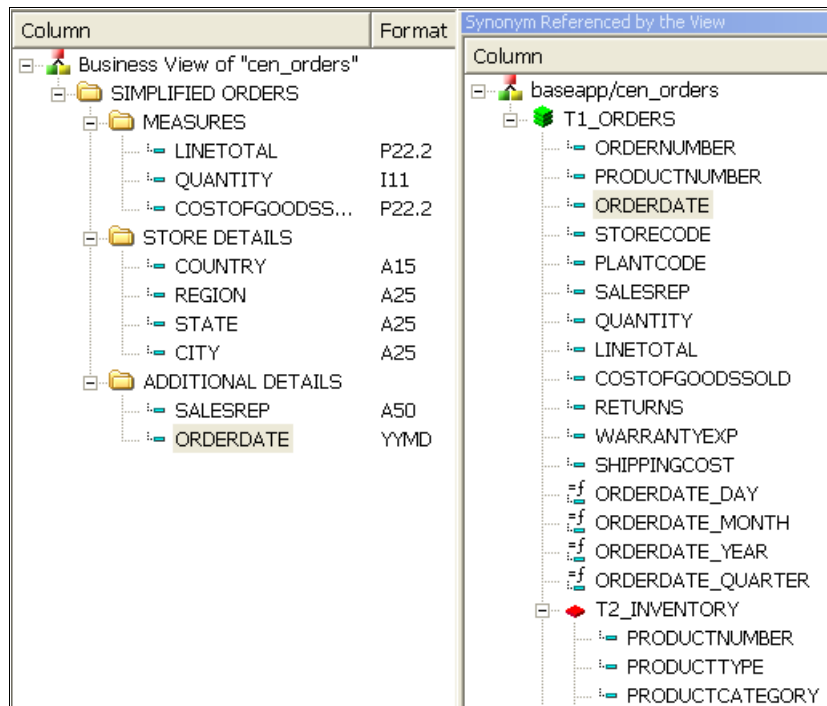


Figure 12-23 Business View

This view is now available to the user who is creating reports just like the original Orders table cluster is.

Tip: If you create a business view for a user, the user can open Power Painter and drag the entire business view to the Power Painter palette, instantly creating a report that uses all the columns in the view. This allows the user to basically create a one-step report.

12.1.6 Impact Analysis

If you are changing a table layout, you can use Impact Analysis to determine the effects of the change on existing reports or business views.

3. To find the reports in the Common Domain that would be affected by a change to the Orders cluster, navigate to **Common Domain**, right-click, and select **Impact Analysis** (Figure 12-24).

Tip: To see all reports that are affected across all folders and all domains, right-click one level higher on Domains.

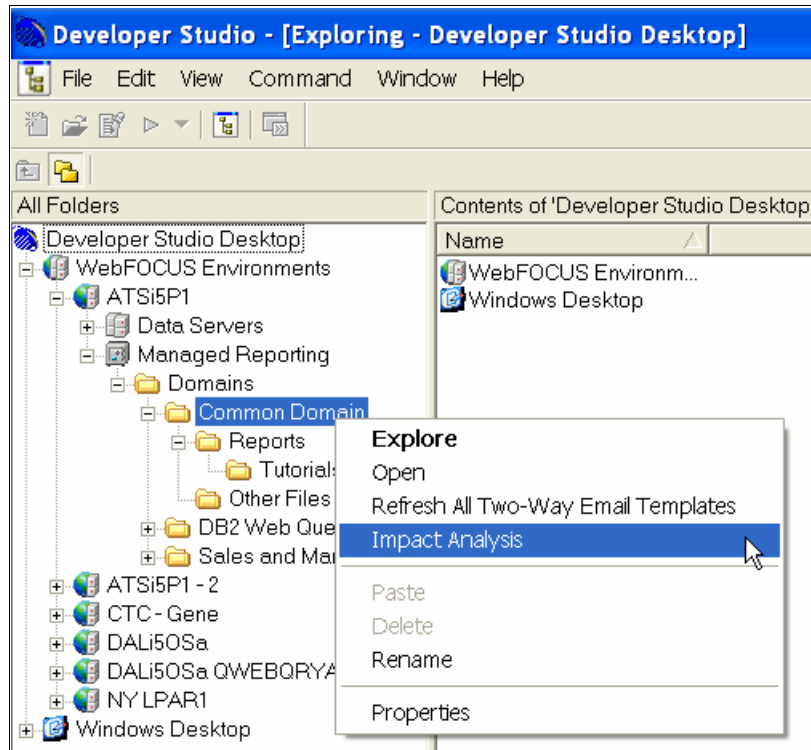


Figure 12-24 Starting Impact Analysis

4. In the Open window (Figure 12-25), double-click **baseapp** to see all our tables and views. In the baseapp folder, select **Cen_orders.mas** and click **Open**.

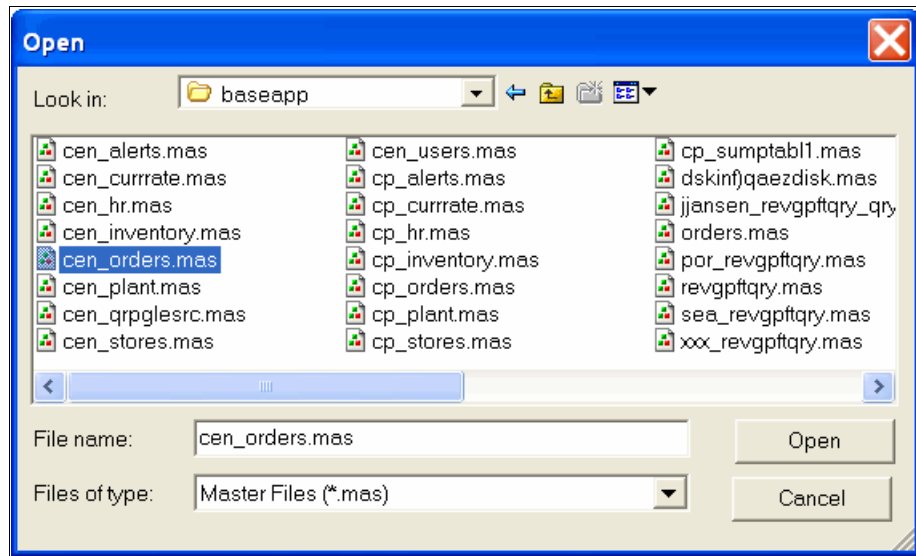


Figure 12-25 Selecting a master file

5. At this point, you can either analyze the impact of modifying individual fields or find any report that references the cen_orders table.

In the Developer Studio - Impact Analysis window (Figure 12-26), click **Analyze** to find all reports that access any column in the cen_orders table.

All reports in the Common Domain that use the Cen_orders table are listed in the bottom pane.

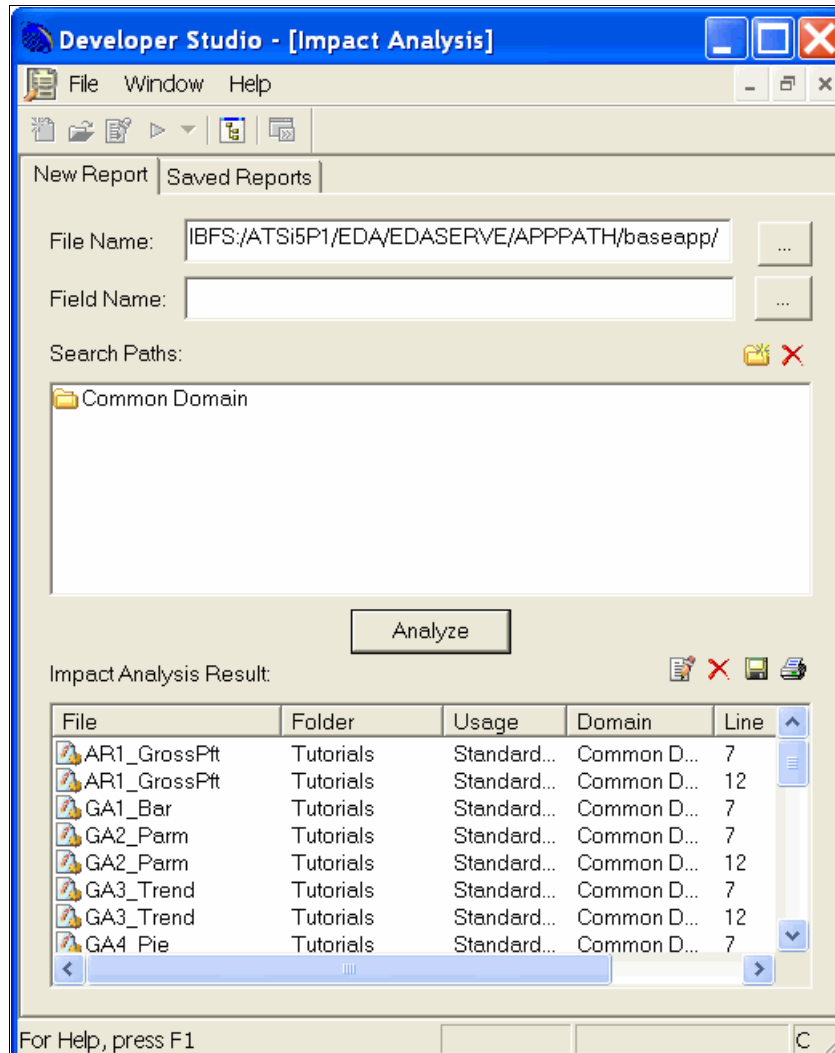


Figure 12-26 Impact analysis results

Tip: To find the business views that use a specific master file or column, open the table in the Synonym Editor and right-click the table or column name and choose **Impact Analysis**.

12.1.7 Data profiling

Developer Workbench provides the capability to profile your data. Profiling allows you to see both the patterns in the actual text characters. In Figure 12-30 on page 290, you can see the four different patterns of postal codes that you have in your data. These patterns might be acceptable because the different patterns represent different countries or they might indicate data quality issues.

Data profiling also shows the minimum and maximum values in a column and identifies ten possible outliers. This is done by showing you the ten highest and ten lowest values in a column as seen in Figure 12-31 on page 291.

1. To start the Synonym Editor, expand the metadata portion of the WebFOCUS Environments. Expand **WebFOCUS Environments** → **your environment** → **Data Servers** → **EDASERVE** → **Application** → **baseapp**.
2. You can profile all the columns in a table by selecting **Data Profiling** or selecting an individual column to profile.

Choosing data profiling at the table level can take a while. Every column in the table is analyzed. The system looks at such items as patterns in the layout of the data in a column, average, minimum, and maximum values, and the length of the data. The number of distinct values is calculated. The number and percent of null values in a column is also calculated. In our example, we profile all the columns in the cen_orders cluster table.

After opening cen_orders in the Synonym Editor, right-click **cen_orders** and select **Data Profiling** (Figure 12-27).

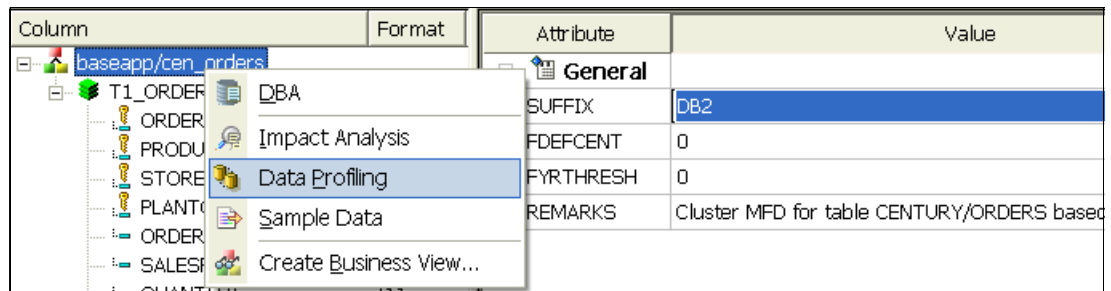


Figure 12-27 Selecting the Data Profiling option

- A summary report is produced like the example shown in Figure 12-28. Notice that the fields in blue contain hyperlinks. Click **POSTALCODE** to see output similar to that shown in Figure 12-30 on page 290.

You can profile an entire table, receive this type of a summary, hyperlinked chart, and drill down into the various options. Another way to obtain these options is to directly profile the field.

	Segment	Name	For...	Distinct Count	Count	Patterns Count	Minimum Value	Maximum Value	Average Value	Minimum Length	Maximum Length	Average Length
1	T1_ORDERS	ORDERNUMBER	A5	5783	32283	2	2B003	9955A	.	5	5	.
2	T1_ORDERS	PRODUCTNUMB...	A4	75	32283	1	1001	5015	.	4	4	.
3	T1_ORDERS	STORECODE	A6	116	32283	1	1003CA	9999CE	.	6	6	.
4	T1_ORDERS	PLANTCODE	A3	6	32283	2	BOS	STL	.	2	3	.
5	T1_ORDERS	ORDERDATE	YYMD	476	32283	.	2003/01/01	2004/12/29	2003/12/09	.	.	.
6	T1_ORDERS	SALESREP	A50	116	32283	55	Adriano Pref...	Wendy Hutson	.	3	23	.
7	T1_ORDERS	QUANTITY	I11	360	32283	.	1	751	70	.	.	.
8	T1_ORDERS	LINETOTAL	P22.2	4358	32283	.	39.00	2028159.00	48382.20	.	.	.
9	T1_ORDERS	COSTOFGOODS...	P22.2	3126	32283	.	15.00	1805600.00	34976.90	.	.	.
10	T1_ORDERS	RETURNS	I11	96	32283	.	0	177	7	.	.	.
11	T1_ORDERS	WARRANTYEXP	P22.2	889	32283	.	.00	270100.00	3535.20	.	.	.
12	T1_ORDERS	SHIPPINGCOST	P22.2	4654	32283	.	.39	42591.34	498.72	.	.	.
13	T1_ORDERS	ORDERDATE_D...	D	31	32283	.	1	31	15	.	.	.
14	T1_ORDERS	ORDERDATE_M...	M	12	32283	.	01	12	05	.	.	.
15	T1_ORDERS	ORDERDATE_Y...	YY	2	32283	.	2003	2004	2003	.	.	.
16	T2_INVENTORY	PRODUCTNUMB...	A4	75	75	1	1001	5015	.	4	4	.
17	T2_INVENTORY	PRODUCTTYPE	A15	5	75	4	Audio	Video	.	5	10	.
18	T2_INVENTORY	PRODUCTCATE...	A30	15	75	14	Amplifiers/Pr...	VCR	.	2	25	.
19	T2_INVENTORY	PRODUCTNAME	A60	75	75	60	Audio/Video ...	7.1-Piece Ho...	.	10	43	.
20	T2_INVENTORY	MODEL	A10	69	75	22	AVR-100	VOE-100	.	5	10	.
21	T2_INVENTORY	QUANTITYINST...	I11	68	75	.	101	9707	1805	.	.	.
22	T2_INVENTORY	PRICE	P13.2	44	75	.	39.00	4599.00	816.06	.	.	.
23	T2_INVENTORY	COST	P13.2	48	75	.	15.00	4000.00	611.93	.	.	.
24	T3_PLANT	PLANTCODE	A3	6	6	2	BOS	STL	.	2	3	.

Options... Refresh Close

Figure 12-28 Data Profiling results at a table level

- Return to the list of columns shown in Figure 12-29. This time we profile an individual column rather than the entire file.

Right-click **POSTALCODE** and select **Data Profiling** → **Statistics**. Notice the different choices you have when profiling a field.

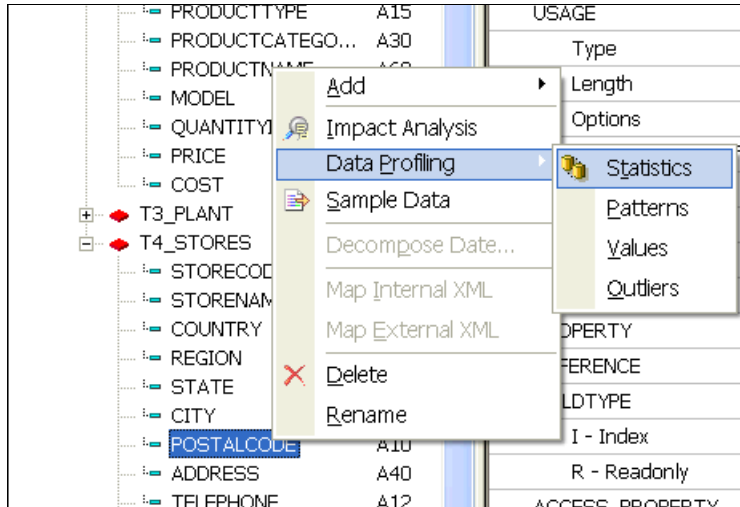


Figure 12-29 Data profiling selection at a field level

In the Data Profiling window (Figure 12-30), you can see either the values and patterns associated with **POSTALCODE** by selecting them from the data profiling submenu or by clicking the appropriate columns in the output report from the statistical analysis.

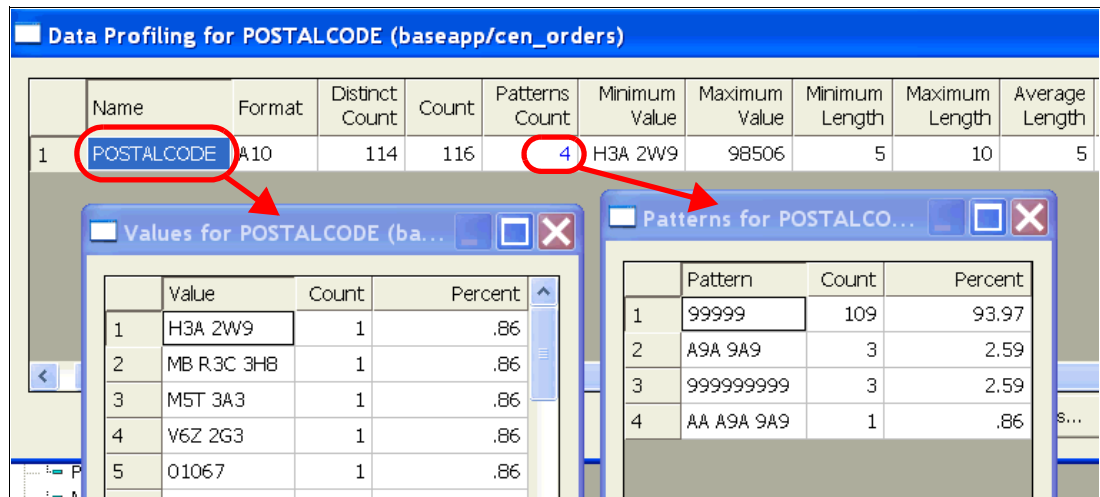


Figure 12-30 Data Profiling results at the field level

- To see data outliers, switch to the **QUANTITYINSTOCK** column. Return to the list of columns (Figure 12-31), right-click **QUANTITYINSTOCK**, and select **Outliers**.

Data outliers are the ten highest and ten lowest values for a column.

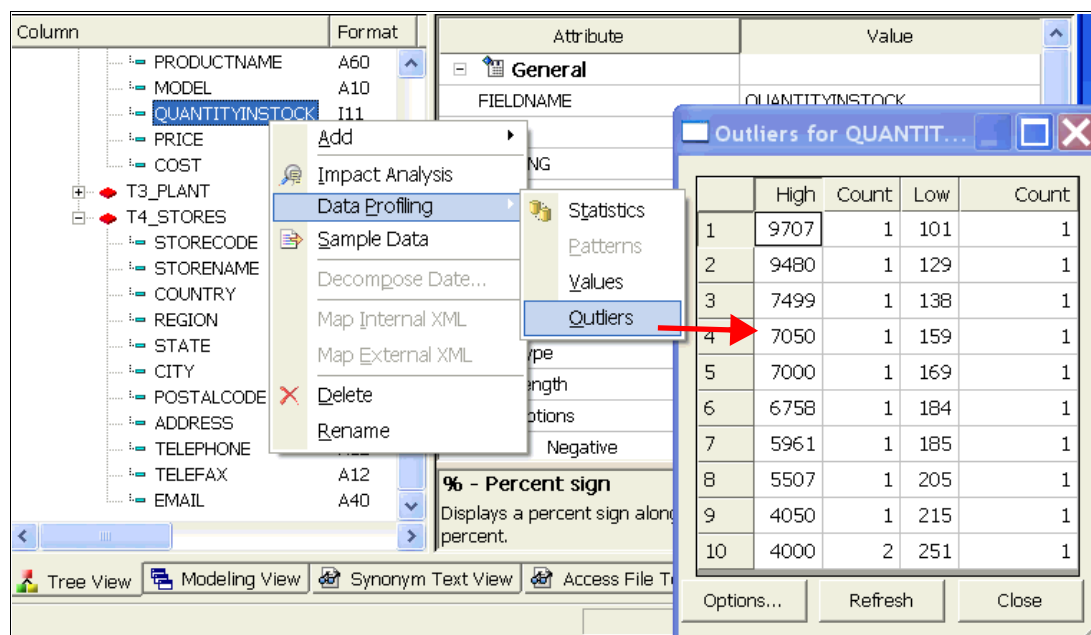


Figure 12-31 Data outliers

12.2 SQL Wizard

In addition to working with the metadata, Developer Workbench includes report development tools. You can directly access Report Assistant and Graph Assistant from within Developer Workbench. One developer's tool that is unique to Developer Workbench is the SQL Wizard.

With the SQL Wizard, we have three different ways to provide SQL statements to DB2 Web Query. In all cases, the SQL statements are run, and the temporary result set becomes the input to a query. You can then modify the output with Report Assistant or Graph Assistant. This is a similar technique to importing Query/400 definitions.

After a report is tailored with one of the GUI tools, it looks like a standard report to the user. When the user chooses to run the report, the SQL statement or SQL stored procedure is executed on the System i environment, and the result set is passed to the reporting procedure for formatting and further modifications. The output is then presented to the user.

Before we start using the SQL Wizard, we need SQL to import into DB2 Web Query:

- Create a file on your PC hard drive called `SQLStmt.sql`. This file should contain the following statement.

```
SELECT INVENTORY."ProductType", INVENTORY."ProductCategory",
SUM(ORDERS."LineTotal") AS Revenue, SUM(ORDERS."CostOfGoodsSold") AS COGS
FROM QWQCENT/ORDERS AS ORDERS, QWQCENT/INVENTORY AS INVENTORY
WHERE INVENTORY."ProductNumber" = ORDERS."ProductNumber"
GROUP BY INVENTORY."ProductType", INVENTORY."ProductCategory"
ORDER BY INVENTORY."ProductType" ASC, INVENTORY."ProductCategory" ASC
```

Running SQL statements from a file requires the file to be in the DB2 Web Query directories in the integrated file system. Although there are many ways to copy your SQL statement there, we use the functionality in Developer Workbench.

2. Make sure that your Windows Desktop is displayed in your Developer Workbench explorer tree. If it is not displayed, see the Explorer options in 9.2, “Setting Developer Workbench default options” on page 198.
3. Open your explorer tree and expand down to **Common Domain** → **Other Files**.
4. Navigating in your Windows Desktop tree, highlight **SQLStmt.sql** and drag it from your PC to Other Files in Common Domain (Figure 12-32).

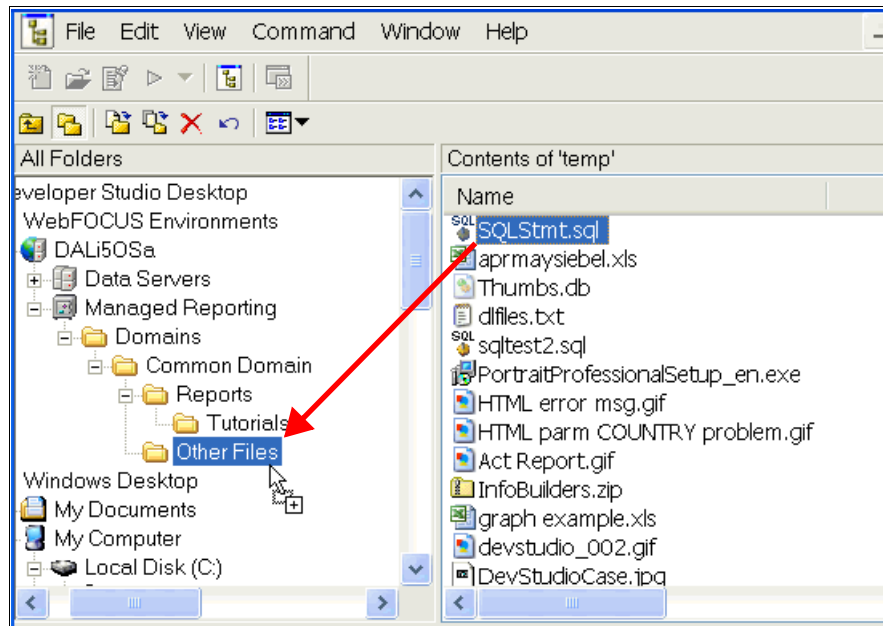


Figure 12-32 Moving PC files to DB2 Web Query directories

- Now that we have the SQL that we want to run in the System i environment, start the DB2 Web Query SQL Wizard. Right-click your **Tutorials** folder, and select **New Procedure** (Figure 12-33).

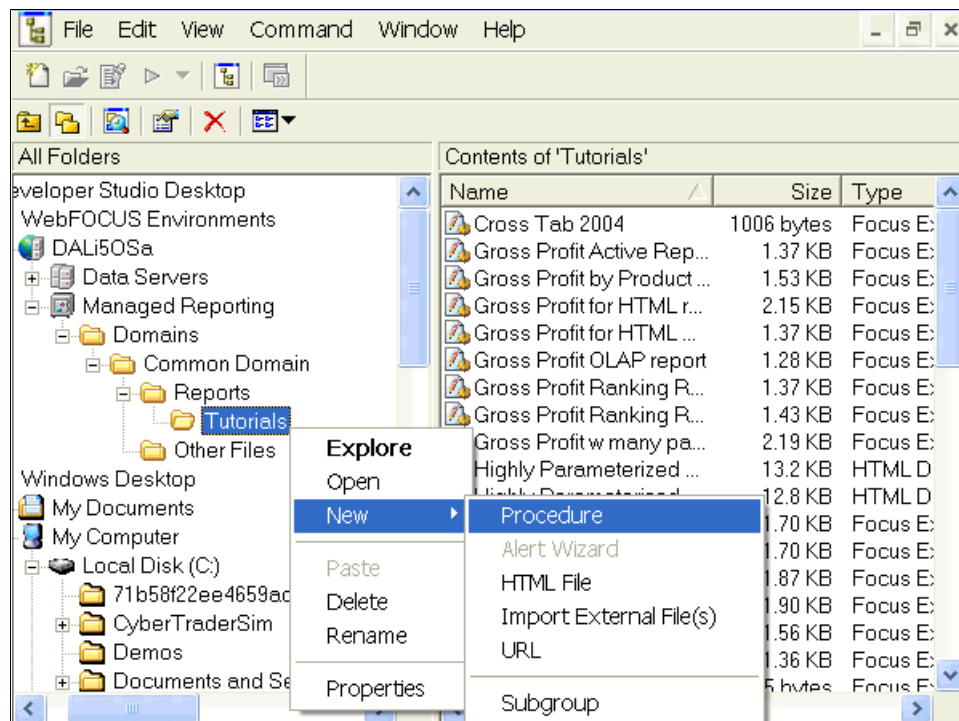


Figure 12-33 Starting the SQL Wizard

- In the Create window (Figure 12-34), for File name, create a file called SQL1. For Create with, select **SQL Report Wizard**. Then click **Create**.

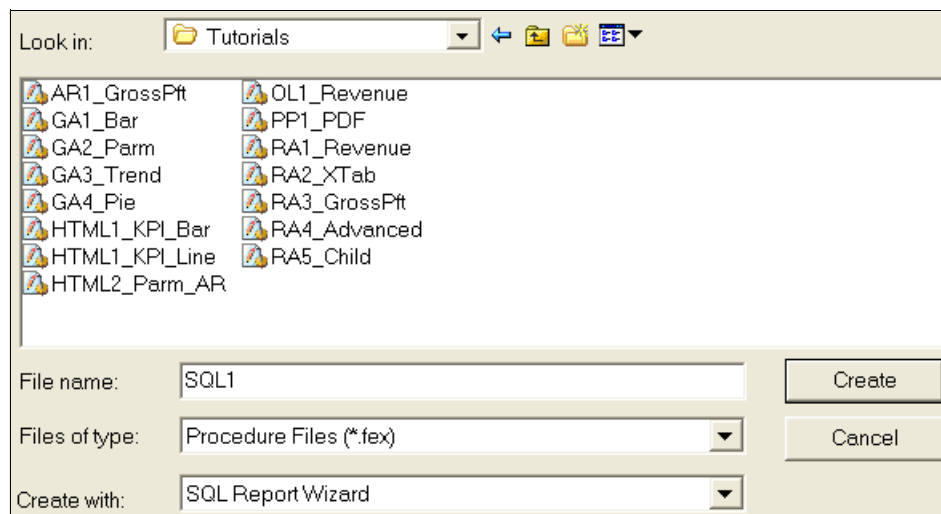


Figure 12-34 Initiating the SQL Report Wizard

Tip: You can go directly to the prompt in Figure 12-34 by using the Quicklinks option of SQL-based reports and graphs.

7. In the Welcome to WebFOCUS SQL Report wizard window (Figure 12-35), you are presented with three options:

- Include an external .sql file

The .sql file must be located in the DB2 Web Query directories in the integrated file system. The SQL statements in the file should generate a single result set.

- Type SQL commands directly

This option allows you to directly enter an SQL command that will create a result set that can be used by DB2 Web Query

- Import an existing .sql file and modify

This is the same as the first option except you are presented with an additional display where you can edit the SQL statements in the file

All options generate an internal, temporary file to hold the results of the SQL execution called SQLOUT.

Select **Included from an external '.sql' file**.

Welcome to the WebFOCUS SQL Report wizard

SQL Reports are a special type of report that is based on an SQL request, rather than a static database description.

There are three ways SQL can be used to create a report:

The SQL commands are stored outside of this focexec

- ☒ Included from an external '.sql' file.

The SQL commands are stored inside this focexec

- ☐ Type SQL statements in this report request
- ☐ Import from an existing '.sql' file.
You will have the opportunity to edit the imported commands.

< Back Next > Cancel Help

Figure 12-35 SQL Report wizard

Tip: SQL stored procedures are not included in the SQL wizard because importing SQL stored procedures is part of the base product. You can do this through the Create Synonym functionality in either the base product or optionally through Developer Workbench.

8. In the next window (Figure 12-36), accept the default options for accessing the database. Click **Create**.

These options tell us how to access the database you will be using.

Select the SQL database engine

What type of SQL database are you reporting from?
(What SQL dialect are you using?)

DB2 cli

Select the connection

What predefined connection do you need to access your data?

*LOCAL

Figure 12-36 Accessing DB2 for System i data

9. As shown in Figure 12-37, for the external SQL file name, click **Browse**. In the Open window, navigate to **Common Domain** → **Other Files**. Select the .sql file **SQLStmt** and click **Open**. Then click **Next**.

Tip: Prior to continuing, you have the option to test the SQL statement against a limited number of records. The output is sent, in default HTML format, to your display.

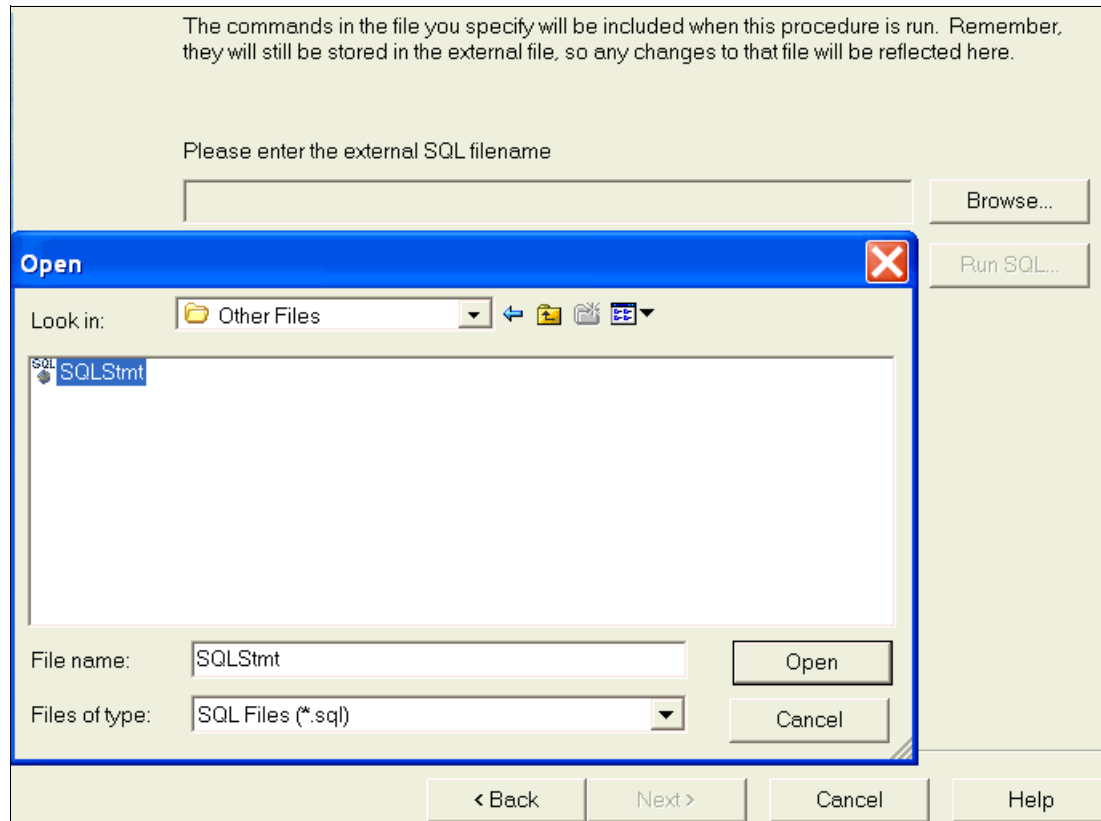


Figure 12-37 Browsing for your .sql file

10. In the Summary window (Figure 12-38), review your options. Then either select **Report Assistant** or **Graph Assistant**. Instead of selecting a table you want to query, you are presented with the output from the SQL. Click **Finish**.

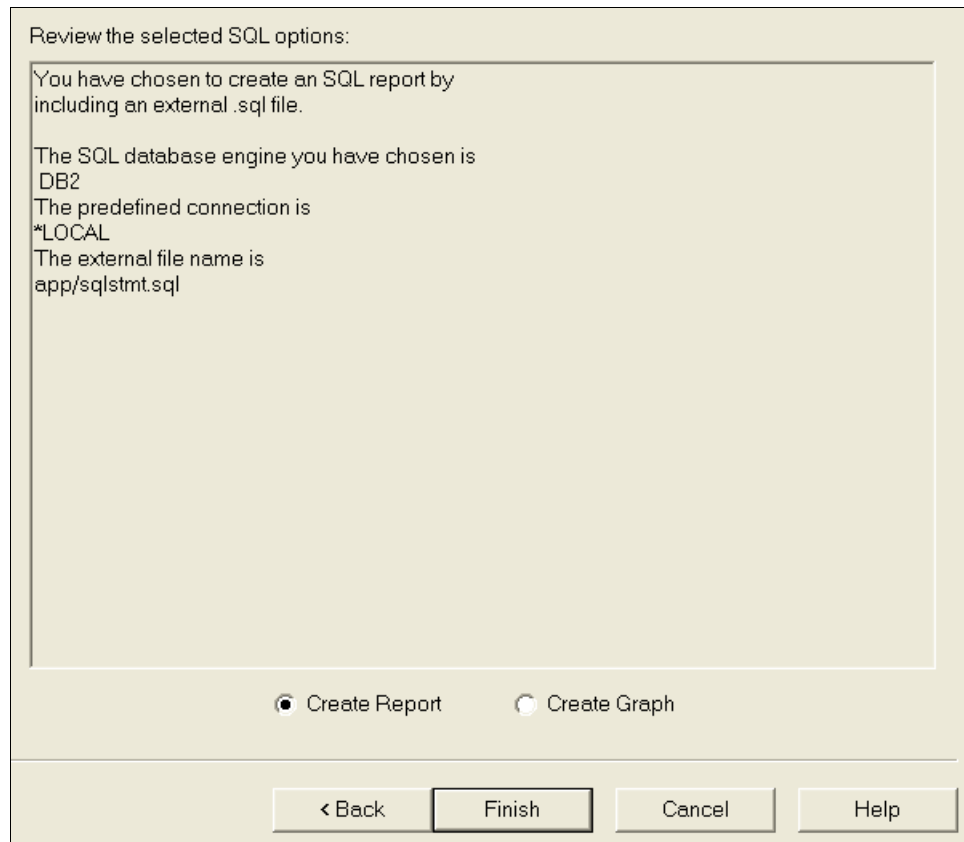


Figure 12-38 SQL Wizard option summary

11. In the window shown in Figure 12-39, notice that the input fields to Report Assistant are the output fields from your SQL execution. You can now choose to format the fields, create additional calculated fields, such as $\text{PROFIT} = \text{REVENUE} - \text{COGS}$, apply style sheets, and do anything else that you normally do to query a table in Report Assistant.

Make some formatting changes, add a heading, and run your report.

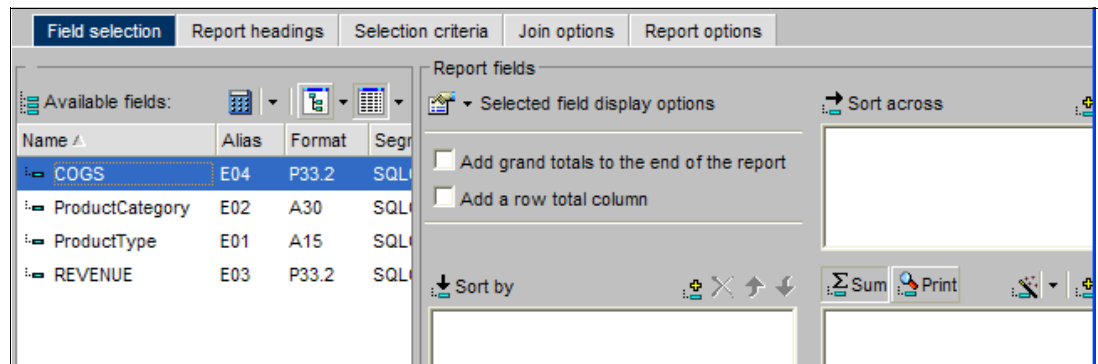


Figure 12-39 Report Assistant with SQL output

12.3 Summary

Between Chapter 9, “Developer Workbench” on page 193, Chapter 10, “Online analytical processing” on page 201, Chapter 11, “HTML Layout Painter” on page 233, and this chapter, you have been introduced to many of the features of Developer Workbench. Developer Workbench is a complete GUI development environment. We highly recommended Developer Workbench to DB2 Web Query developers.

You have now completed all the tutorials for DB2 Web Query. Now it is time for you to start using this query and reporting tool against your own data. We have tried to introduce you to the more common features within this product. There are many additional features that we have not touched upon. Feel free to experiment and remember that you can always turn to the help text if you need assistance. The help text accessed in Developer Workbench is much more detailed and complete than the basic help text that is available for the other Web components.



Part 4

Miscellaneous or additional topics

This part includes the following chapters:

- ▶ Chapter 13, “Using DB2 Web Query to run existing Query/400 reports” on page 301
- ▶ Chapter 14, “Performance considerations” on page 309
- ▶ Chapter 15, “Frequently Asked Questions” on page 341



Using DB2 Web Query to run existing Query/400 reports

In this chapter, we discuss the similarities and differences between Query/400 and DB2 Web Query. We also discuss common uses for Query/400 and how DB2 Web Query can maintain that same functionality. In addition, we show how to create metadata from your QRYDFN objects so DB2 Web Query can run your existing queries.

13.1 Query/400: A reliable reporting tool

By now, you understand that DB2 Web Query is the intended replacement for Query/400. This does not imply that familiar commands such as WRKQRY and RUNQRY are going away, but rather product Query/400 is going away. You can still use older, yet robust, technologies that have stood the test of time and are useful to your business. It is likely that Query/400 falls into this category and has been a reliable reporting tool that your users have become easily adept with. Query/400 appeals to users because it is easy to learn and does not require programming skills or knowledge in SQL. Also since IBM development has not made recent changes to the Query/400 code, the user interface has remained familiar and code defects are practically non-existent.

DB2 Web Query has the ability to run your existing Query/400 reports without you changing or running a conversion process on the QRYDFN object. You only need to create metadata on the QRYDFN object like you do for a DB2 table. Then DB2 Web Query submits the actual RUNQRY command to the server and displays the results to your browser. Just as RUNQRY has the three output options of display, spool file, or database file, DB2 Web Query also performs the output option that is defined in the QRYDFN.

As discussed in Chapter 3, “DB2 Web Query fundamentals” on page 25, DB2 Web Query requires special metadata to read from table objects. This is also true of QRYDFN objects. The process of creating metadata on QRYDFN objects is quite similar to the process for tables. After the metadata is created, DB2 Web Query treats table and QRYDFN metadata in quite the same way.

13.2 DB2 Web Query versus Query/400: Function similarities

In the following list, we summarize how DB2 Web Query and Query/400 have the same functionality:

- ▶ *Report appearance:* Based solely on report appearance, a basic user is unable to tell if the report that is running is based on a DB2 object or a QRYDFN object. The information and appearance of both final reports based on QRY/400 metadata versus DB2 file metadata are identical.
- ▶ *Spooled file output:* If your QRYDFN object is defined with the output option of Printer, DB2 Web Query sends the output to a spooled file.
- ▶ *Database file output:* If your QRYDFN object is defined with the output option of Database file, when DB2 Web Query runs that report, it sends the output to a Database file.
- ▶ *Batch capabilities:* It is common to submit a CL program to batch that performs several RUNQRY commands. DB2 Web Query can also perform this functionality. The eventual goal in some shops might be to completely replace their QRYDFN objects with DB2 Web Query reports. With this in mind, DB2 Web Query has a thread-safe command that allows all your reports developed in DB2 Web Query to be used in your batch jobs.
- ▶ *Right Join:* WRKQRY does not support Right Join and neither does DB2 Web Query.

Just as DB2 Web Query supports Query/400 functions, likewise the reverse is true. With the exception of what we discussed in Chapter 5, “Report Assistant” on page 63, all Report Assistant functions can be used on Query/400 metadata. Recall that DB2 Web Query sees the Query/400 result set much like a table.

13.3 DB2 Web Query versus Query/400: Function differences

In this section, we discuss the functionality that Query/400 has that DB2 Web Query does not have. In all cases, the limitation is explainable and a workaround is provided.

Parameter passing

It is possible to create a QRYDFN object that can accept parameters at run time. This is done by using a variable in the selection clause. Figure 13-1 shows how this is used. Then the QRYDFN object is called by using the STRQMQRy ALWQRyDFN (*YES) command, which prompts the user for parameters.

AND/OR	Field	Test	Value (Field, Number, 'Characters', or ...)
	CITY	EQ	:MYCITY

Figure 13-1 QRYDFN with parameters

As noted earlier in this chapter, DB2 Web Query sends the RUNQRy command to the server. It does not send any version of STRQMQRy. This limitation means that QRY/400 queries that were coded for use as STRQMQRy ALWQRyDFN(*YES) to allow parameter input must instead use the DB2 Web Query parameter passing. First, you must change your QRYDFN object to not accept parameters. After you create the metadata, you can use Report Assistant to code the parameters into the selection criteria. See 5.4.1, “Parameterized reports” on page 98, for creating a parameterized report.

Using output files for the next query

It is common to use RUNQRy outfile results for use in the next RUNQRy. It is still possible to do this “chaining” in DB2 Web Query, although you must first create metadata on the intermediate output files. The metadata is hardcoded only with a library and file name; it does not check the format-level identifier of the file as some programs do.

For example, library QGPL is used to hold the work files. You must run all the reports and generate the outfiles to QGPL. Do not delete the QGPL files. Then you create the metadata only once on the QGPL work files and create reports from the work files. When the job is done, the work files are deleted. The next time the job runs, the work files are created again. As long as the files are created with the same fields and format as before, the previously created metadata successfully runs the report. DB2 Web Query does not know that the underlying files have since been deleted and recreated.

Commonly the library QTEMP is used to store these intermediate work files. The limitation of DB2 Web Query is that you cannot create metadata on any object in QTEMP. Therefore, the intermediate file must be in a different library; then the metadata can be created. If you cannot change your work files to be in the QTEMP library, you will be unable to do query chaining. If you change your work files from QTEMP to another library, be mindful to change your program to delete the work files.

In summary, metadata remains valid even if the underlying object is deleted and recreated. However, if the underlying object is changed, then the metadata is stale and must be recreated.

13.4 Creating metadata from Query/400 objects

Just as DB2 Web Query requires metadata to read from DB2 files, it also requires metadata to read from Query/400. When creating metadata on QRYDFN objects, you must remember to complete a few extra windows; otherwise the metadata creation will not be complete.

As noted earlier, if you change the format of your underlying files, you must recreate the metadata on those files. If you change the QRYDFN on the server side, you must recreate the metadata on that QRYDFN.

The process of creating metadata on a Query/400 runs the query at that time. This is because it is gathering information about the files and columns that were used in the query output. For this reason, the processing of creating metadata on a Query/400 object can be somewhat longer than on a DB2 table.

DB2 Web Query asks for the library name where your Query/400 objects are stored. You can choose to create metadata for all queries in a library in a single execution. In the following steps, we explain the process when creating Metadata for Query/400. There are two additional windows for creating metadata on a Query/400 object versus on a DB2 object. Remember to click the Next button until you no longer see one.

1. In the left pane of the Data Adapters page (Figure 13-2), select the adapter to use. In this case, click the **Query iSeries adapter** and select **Create Synonym**.

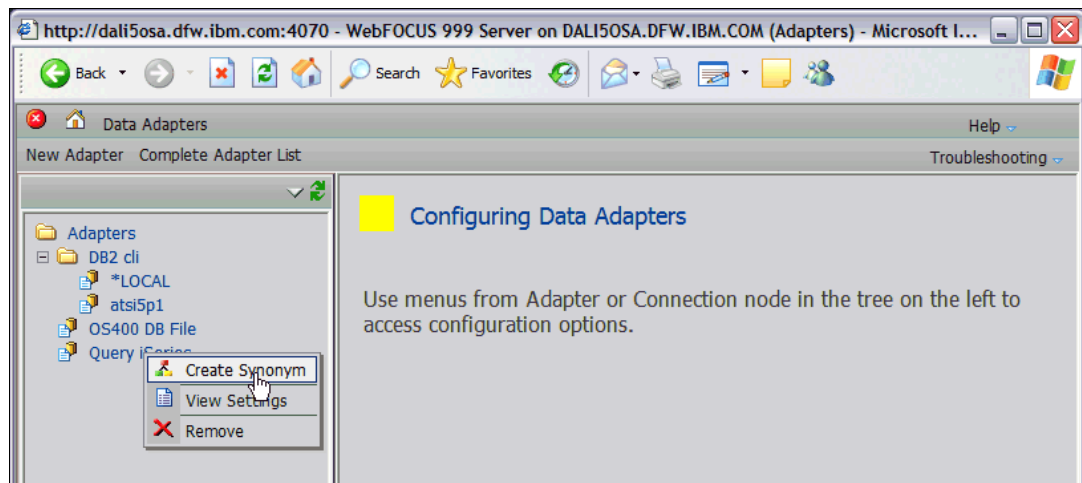


Figure 13-2 Selecting the data adapter

2. In the Query Synonym for Query iSeries pane (Figure 13-3), enter the library where the QRYDFN object resides. In this case, we enter the RMMASON library. Click **Submit**.

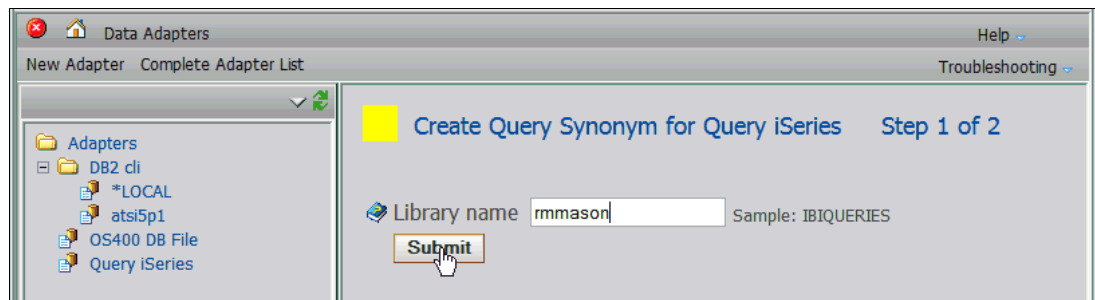


Figure 13-3 Creating a synonym

3. In the Select Synonym candidates for Query iSeries pane (Figure 13-4), complete these steps:
 - a. Select the Query/400 object on which you want to create metadata.
 - b. If you previously created a synonym on this object and want to replace it, select **Overwrite existing synonym**.
 - c. In this case, this is the first time for creating metadata on QRYDFN in this library. Therefore, we select **Default synonym name**. By selecting this option, every QRYDFN in the entire library is selected.

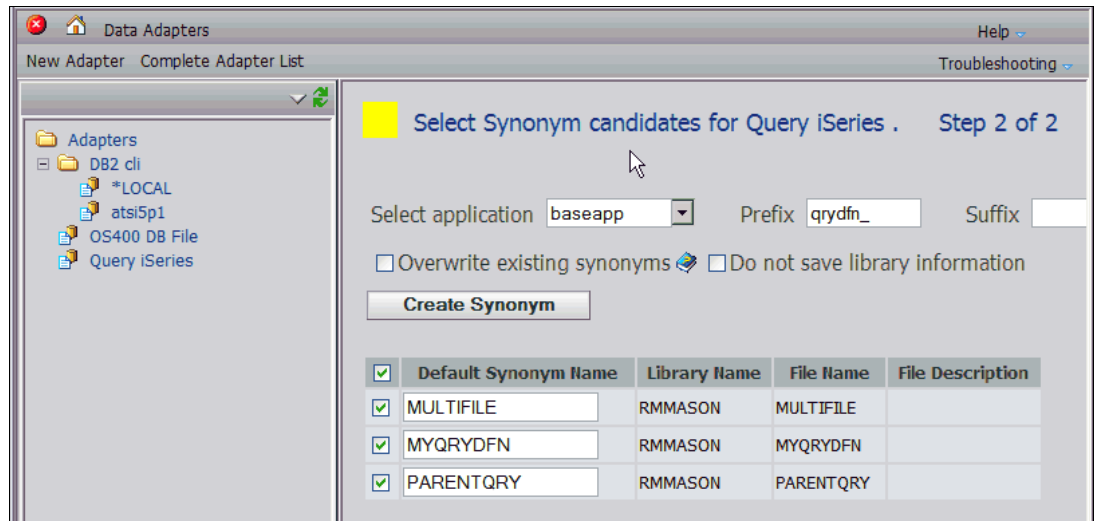


Figure 13-4 List of QRYDFN objects

- d. You see a warning message like the one in Figure 13-5, which says that the synonym creation process can take a long time. Click **OK** to indicate that you want to continue.

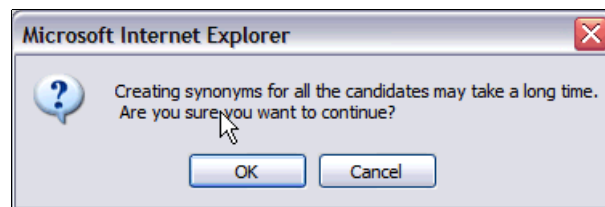


Figure 13-5 Warning message - Creation of synonyms

- e. Click **Create Synonym**.

4. In the Create Synonym for Query iSeries pane (Figure 13-6), if synonyms creation was successful, you see a status of “Create successfully”. Click **Next**.

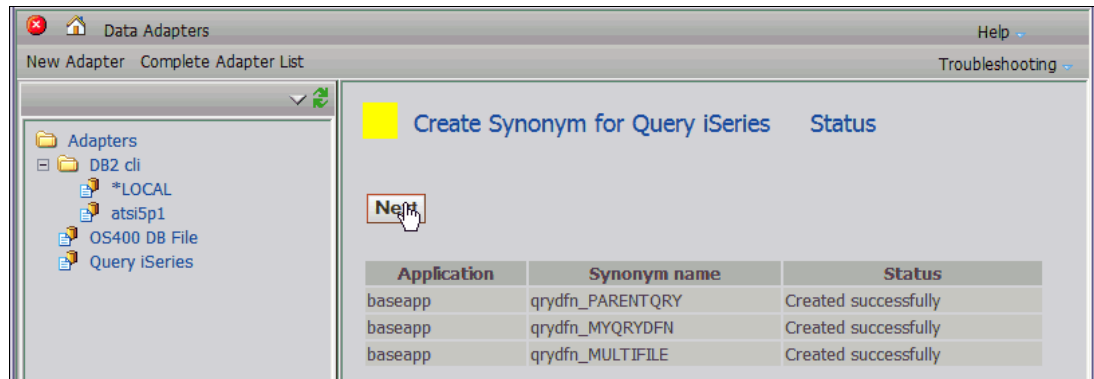


Figure 13-6 Created successfully status message

5. In the Create Synonym for Query/400 File: Save Reports pane (Figure 13-7), notice the Report Name column. Click **Next**.

This is the last pane on which you must click Next. DB2 Web Query creates a report from your QRYDFN.

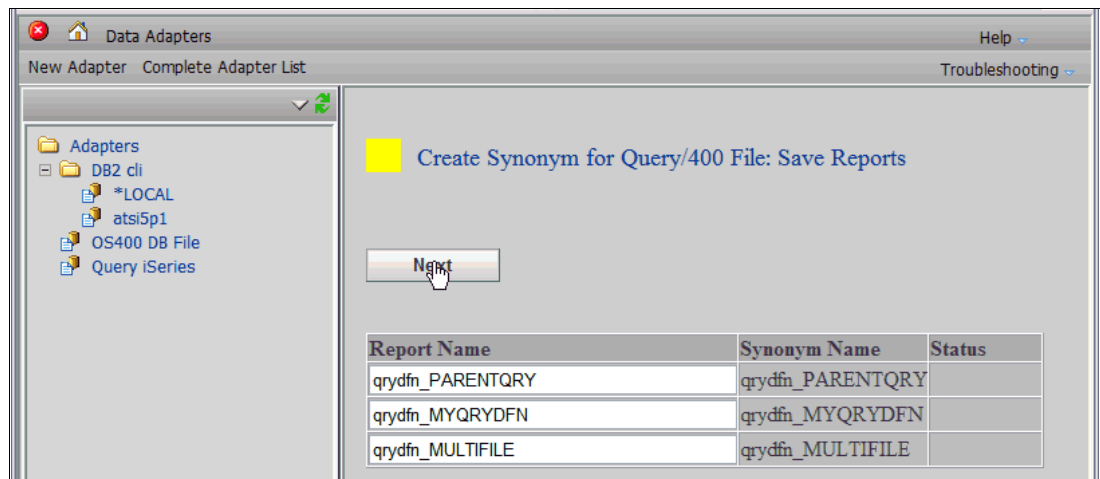


Figure 13-7 Final confirmation window - Creation Synonym for Query/400

6. In the Create Synonym for Query/400 File: Save Reports Status pane (Figure 13-8), notice that the status indicates “Created successfully” for the Query/400 metadata creation. Close this window.

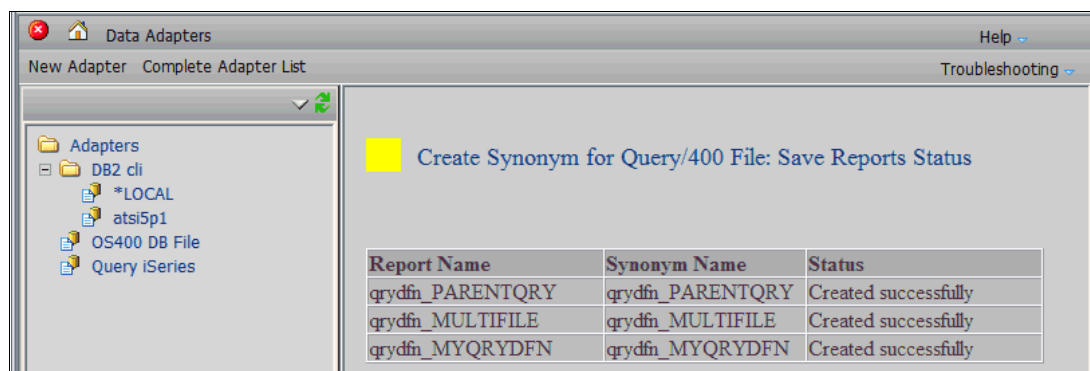


Figure 13-8 Created successfully message - Conversion from Query/400 to DB2 Web Query

Now if you go back into your DB2 Web Query domain, notice that the metadata is already created as a report. You can click this report to run it or right-click it for the editing options.

Options when creating metadata: Prefix and suffix

The guidelines for adding a prefix and suffix to metadata are the same as outlined in Chapter 3, “DB2 Web Query fundamentals” on page 25. Keep in mind that after metadata creation, you will be unable to tell if the underlying object is a DB2 table or a QRYDFN. You might want to add a prefix or suffix that can help you to identify that the object being queried on the server side is a QRYDFN.

13.5 Using DB2 Web Query to edit a QRYDFN

After you create metadata, DB2 Web Query understands the metadata as a result set. You can use DB2 Web Query to edit, sort, and filter the report, but you can only work within that result set. You cannot add new columns to the report, even if the column is from a table that is used in the report. DB2 Web Query, in effect, sees the Query/400 result set as any other table.

For example, in Chapter 5, “Report Assistant” on page 63, we learn the difference between compute and define. Since compute is data manipulation done on the client side, you can still use compute on your Query/400 reports. However the define operation is data manipulation performed on the server side. Therefore, you are unable to do this when editing a Query/400 object. Currently bad data is returned and might change into an error message instead.

If you edit the report from DB2 Web Query, those changes are not reflected in the QRYDFN object. When editing a Query/400 report that contains a join, you can add a new join with DB2 Web Query. However, you will be unable to edit the joins already defined in Query/400.



Performance considerations

Minimizing the processing time for report requests and maintaining the accuracy and integrity of the report data are major goals for most organizations. In this chapter, we discuss DB2 Web Query performance considerations and recommendations. We include a case study that is intended to demonstrate steps that you can take to tune a DB2 Web Query report.

14.1 Performance basics

When the objective is to obtain optimal report performance, keep in mind the following DB2 Web Query characteristics regarding how it approaches performance optimization:

- ▶ When a DB2 Web Query report is run, the tasks of row selection, joining, ordering, and aggregating must be performed. These tasks can be carried out by the DB2 for i5/OS database engine, DB2 Web Query, or a combination of both.
- ▶ DB2 Web Query attempts to translate the source code of a DB2 Web Query report into equivalent SQL statements. When the report is run, the SQL statement is submitted to the database engine for processing.

Attention: SQL translation does not occur for reports that go through the Query/400 or DB Heritage Files adapter. Those adapters use different methods to access the data.

- ▶ Like most other relational databases management systems (RDBMS), the DB2 for i5/OS database engine knows its own data. Its optimizer component uses complex costing algorithms to determine the ideal access plan, which indexes to use, and so on, so that it can retrieve the data in the most efficient manner. To aid in its decision making processes, it maintains various forms of statistics about the data. In addition, much of the database code lies below the Machine Interface (MI) layer of i5/OS, which is an attribute that can yield great efficiencies during the query optimization and execution steps.
- ▶ For accomplishing all processing tasks in an efficient manner, the database engine should be preferred over DB2 Web Query. Therefore, the goal of DB2 Web Query optimization is to push as much of this processing, decision making, and execution as possible down to the database level.

If a request is not optimized, DB2 Web Query is forced to perform tasks such as joining, sorting, and aggregating. In these cases, the database engine retrieves all the rows that are necessary for the request based on the translation. DB2 Web Query then completes that processing necessary to prepare the data for the report. This can have the following results:

- ▶ Higher processing costs for both the database engine and DB2 Web Query
- ▶ Potentially higher network communication costs since larger-than-necessary result sets are moved around the network
- ▶ Higher personnel costs since time is wasted waiting for inefficient queries to execute and return the requested data

14.2 DB2 CLI adapter performance

In this section, we discuss performance considerations that are specific to the DB2 call-level interface (CLI) adapter.

14.2.1 Report request process flow

As a foundation for understanding DB2 Web Query performance factors, you must first understand the overall process flow when reporting from relational data. It is important to comprehend this process flow so that you can see where their different phases of optimization occur when a report request is made and how you can influence them.

The process of running a DB2 Web Query request consists of two phases:

- ▶ The execution phase
- ▶ The report production phase

Figure 14-1 illustrates the flow of the request through these two phases.

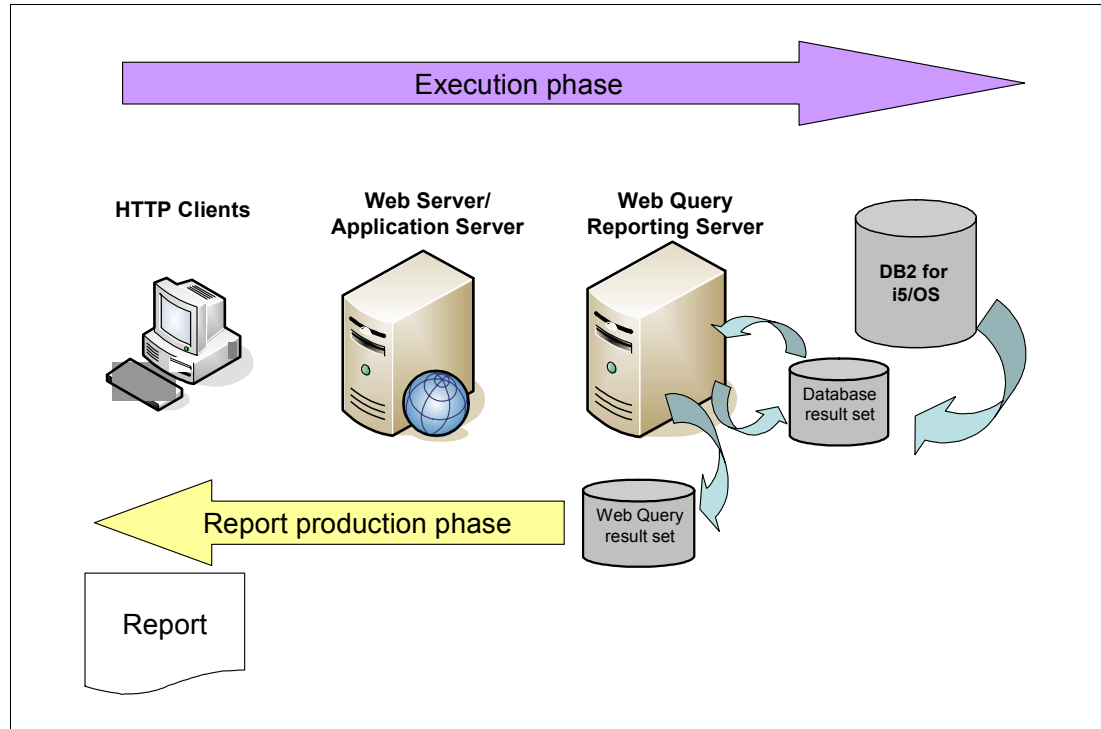


Figure 14-1 Flow of DB2 Web Query report run request

Report execution phase

The following steps occur during the execution phase of a report request:

1. The Web browser client sends the request to the Web server or application server for processing.
2. The Web server or application server routes the request to the DB2 Web Query Reporting Server.
3. The Reporting Server component handles the following tasks during this phase:
 - a. Reads and parses the DB2 Web Query metadata
 - b. Parses the procedure (checks the syntax of the source code)
 - c. Sends a DESCRIBE request to DB2 for i5/OS, so that the Reporting Server can create metadata for the result set that will be returned
 - d. Passes the procedure to the data adapter for processing
4. The DB2 Web Query data adapter component handles the following tasks during this phase:
 - a. Analyzes the DB2 Web Query master file for the specific SQL module and dialect to use, and to retrieve the SQL column names
 - b. Analyzes the DB2 Web Query access file for the specific table to access, and which connection to use

- c. Translates (optimizes) the DB2 Web Query request to the appropriate SQL statement or statements
- d. Passes the SELECT statement or statements to the DB2 for i5/OS database engine
- 5. The database engine component handles the following tasks during this phase:
 - a. Analyzes and optimizes the SQL statement or statements
 - b. Chooses the appropriate access path and retrieval method
 - c. Retrieves the data
 - d. Creates the database result set
- 6. The report Execution phase ends.

Report production phase

At the completion of the report execution phase, the report production phase begins. During the report production phase, the database engine component sends a row and an SQL status code to the Reporting Server.

Note: The database engine is now interfacing with the database result set and not the native relational data on DB2 for i5/OS.

The Reporting Server component handles the following tasks during this phase:

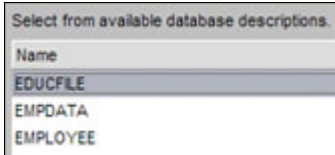
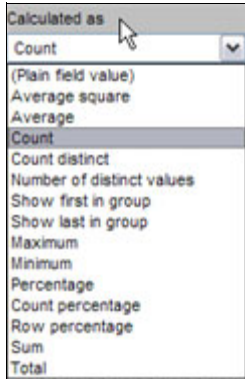
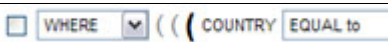
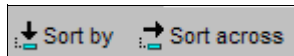
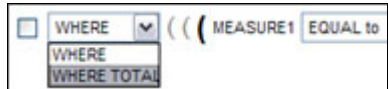
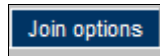
- 1. Reads the row from the database result set and processes any remaining actions on that row (IF/WHERE, DEFINEs, and so on that were not translated by the database engine).
- 2. Converts nonstandard data into DB2 Web Query format, making it available to DB2 Web Query.
- 3. Puts the valid row into the DB2 Web Query result set.
- 4. Asks the database engine for the next row (FETCH).
- 5. Repeats tasks 1 – 4 (loop) until the end of the database result set (SQL status code +100).
- 6. Processes the DB2 Web Query result set, applies style sheets, and formats the report output as requested by the instructions in the procedure.
- 7. Displays the report via a browser or native program based on the type of output produced (for example, Adobe or Excel).

14.2.2 Adapter processing and optimization

When reporting from relational data sources, the data adapter attempts to translate the source code of a DB2 Web Query procedure into equivalent SQL statements. *Adapter optimization* is the degree to which a DB2 Web Query request is translated to pure SQL statements to be handled by the database engine. Maximizing the adapter optimization should be a major goal of any DB2 Web Query application. If all the commands in a DB2 Web Query report request are translated into the equivalent operation in SQL, then all joining, sorting or aggregating functions are handled by the database engine. When this occurs, the request is considered (from a DB2 Web Query perspective) to be totally optimized.

Table 14-1 displays the DB2 Web Query operations and the SQL equivalent.

Table 14-1 DB2 Web Query operations and SQL equivalent

Operation	SQL equivalent
Retrieve rows for columns specified	SELECT SELECT DISTINCT(...)
Identify the table from which to report 	FROM
Simple aggregation functions 	SUM, COUNT, AVG, MIN, MAX
Retrieve specific rows 	WHERE
Ordering rows 	ORDER BY
Retrieve specific aggregated rows 	HAVING
Creating logical table structures (joins) 	FROM...WHERE...

Selection and projection

Two important processes that occur during the SQL translation step of the execution phase are selection and projection. Each is discussed in this section.

► Selection

Selection is the process of retrieving the table rows that meet the request criteria.

Selection is translated to various predicates of the SQL WHERE clause *except* those expressions listed in Table 14-2.

Table 14-2 Selection disablers

Selection disabler	Description
Non-translatable DEFINE fields	Certain DEFINE expressions can be translated to SQL as part of aggregation or record selection operations. In some circumstances, they can be used as part of a define-based join. The following examples are of translatable DEFINE expressions: ► For selection, the DEFINE expression must be an arithmetic valued expression, a character string valued expression, or a logical expression. ► For aggregation, the DEFINE expression must be an arithmetic or character string valued (COUNT) expression. ► These expressions are based on what is built (for example, on the right side of the equal sign) in the DEFINE expression, <i>not</i> on the format of the field that is being defined. Table 14-3 provides more information about the DEFINE field expressions that can be translated to SQL.
Non-translatable DATE fields	DATE fields with formats other than YMD or YYMD
Spanned DEFINE fields	DEFINE fields that span more than one segment in a joined structure

Table 14-3 lists the DEFINE expressions that can be translated to SQL.

Table 14-3 DEFINE expressions that can be translated to SQL

Expression®	SQL translatable conditions
Arithmetic expressions NEWSAL/D12.2 = ((CURR_SAL + OTIME_SAL) x 1.1) - 100;	► Real-field operands of numeric data types (I, P, D, F) ► Numeric constants ► Arithmetic operators (*, /, +, -) ► Subtraction of one DATE field from another ► DEFINE-field operands satisfying any of the above
Character string expressions FORMAL_NAME/A36 = LAST_NAME ', ' FIRSTNAME;	► Real-field operands of alphanumeric data types. (TEXT field formats are not supported with DEFINE expressions) ► String constants ► String concatenation operators ► DEFINE-field operands satisfying any of the above
Logical expressions (Expressions that are evaluated as true, 1, or false, 0.) SALES_FLAG/I1 = (DIV_CODE EQ 'SALES') OR (COMMISSION GT 0); QUOTA_CLUB/I1 = (SALES_FLAG) AND (UNITS_SOLD GT 100);	► Real-field operands of any DB2 Web Query-supported data type (including DATE fields) ► Constants with same data type as field(s) in the predicate ► Relational operators (EQ, NE, GT, LT, GE, LE) ► Logical operators (AND, OR, NOT) ► Arithmetic or character string expression operands (above) ► DEFINE-field operands satisfying any of the above

For more information about creating DEFINE fields, refer to “Defining and computing fields” on page 81.

► **Projection**

Projection is the process of retrieving the table columns that meet the request criteria. Projection is translated to be objects of the SELECT statement as follows:

- Columns referenced in PRINT, SUM or COUNT commands.
- Columns used as objects in JOIN or DEFINE operations.
- PRINT * and SEG.fieldname return all columns in the Master File only.

Note: SELECT * is never generated by DB2 Web Query.

It is important to understand the DB2 Web Query selection and projection operations because they are both processes that reduce the volume of data that is returned from the database, which helps to improve performance, efficiency, and report response time.

Optimization hierarchy

In addition to individual operations being optimized, a progressive optimization hierarchy affects the SQL that is being generated. Figure 14-2 shows the order of adapter optimization.

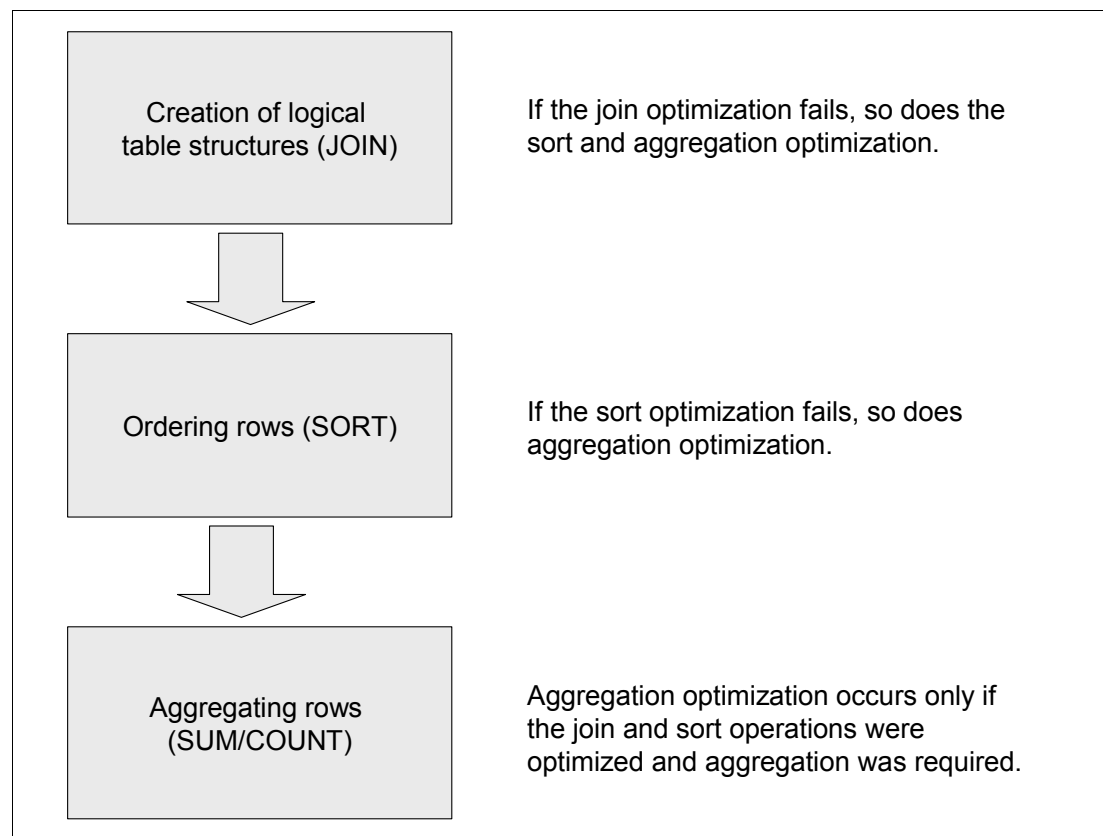


Figure 14-2 DB2 Web Query adapter optimization order

Translation to SQL: Regardless of the hierarchy in Figure 14-2, the adapter always tries to translate row selection (WHERE) and projection operations to their SQL equivalents.

This hierarchy is mentioned because it is important for you to know what to focus on first. If your report's ordering or aggregating is not being performed by the database engine and you are unable to determine why, it might be because join optimization is failing. Therefore, if you have a poorly performing report that is joining two or more tables, your analysis should begin with join optimization.

Join optimization

A join operation is a complex function that requires special attention in order to achieve good performance. Again, it must be stressed that the goal is to allow the database engine to perform as much of the optimization and processing as possible. Due to their potential complexity, this is especially true for join operations.

If the join is processed by the database engine, the following steps occur:

1. The DB2 Web Query request is optimized and is translated to a single SQL select statement.
2. The database engine retrieves the data from the tables specified in the select statement.
3. The database engine merges the rows using its chosen join implementation
4. Screening conditions are applied, if any.
5. The columns in the request are applied.
6. Any column function values or expressions are calculated.
7. One result set is produced.

Table 14-4 illustrates the strengths of DB2 Web Query as well with the corresponding strengths of DB2 for i5/OS.

Table 14-4 Quick guide to strengths of DB2 Web Query and DB2 for i5/OS

DB2 Web Query strengths	DB2 for i5/OS strengths
DB2 for i5/OS federated joins	Joining on local tables (SELECT...FROM...WHERE)
Complex calculations	Row and Column selection (SELECT...WHERE)
Sophisticated formatting	Sorting (SELECT...ORDER BY)
Generate graphs	Aggregation (SELECT...GROUP BY)

Referential integrity and constraint awareness: As of V5R3, the SQL Query Engine also has referential integrity and constraint awareness. This means that the optimizer is capable of using the referential integrity conditions to rewrite the queries to eliminate unnecessary join combinations. This can result in significant performance gains for queries with complex joining and is another compelling reason to try to achieve full DB2 Web Query optimization.

DB2 for i5/OS usually handles the join if all the DB2 Web Query commands (where possible) are translated to their SQL equivalent. However, because of the processing hierarchy, sorting and aggregation can still fail.

Conditions that prevent full DB2 Web Query optimization

After a query request is submitted, there are several conditions that cause the adapter to disable the optimization of the join operation. These *join disablers* are shown in Table 14-5. You must eliminate these conditions if you want full DB2 Web Query optimization to occur.

Table 14-5 Join disablers

Join disabler	Description
DEFINE based JOINS (only if the DEFINE expression is not translatable)	A DB2 Web Query define-based join allows joining a cross-reference file to a host file field that was created by a DEFINE statement. As mentioned previously in “Selection and projection” on page 314, there are cases in which DEFINE fields are not translated to SQL. When this happens, and that DEFINE field is used as a join field, the join is not translated either.
DEFINE fields that span more than one segment in a joined structure	Joining tables based on DEFINE fields prevents the database engine from processing the join.
Multiplicative effect	An aggregation at any level other than the lowest level of the JOINed structure <i>or</i> a child table whose foreign key does not totally cover its primary key (causing the parent table rows to be duplicated). For more information about the multiplicative effect, refer to “Optimization and avoiding the multiplicative effect” on page 317.
Federated joins	Joining DB2 for i/OS tables on different systems, partitions, or independent auxiliary storage pools (IASPs) is carried out by DB2 for DB2 Web Query and not the database engine.

When you create a join connection (via the join tab) between tables, DB2 Web Query might choose to generate and run multiple SQL statements to compute the correct output for your report. In some cases, DB2 Web Query can reduce these multiple SQL statements to a single SQL statement thereby improving DB2 Web Query’s performance considerably. In particular, when the joined in table (called the *target table* on the right of the join panel) is joined via a unique join, DB2 Web Query can process the entire report in one underlying SQL query. A *unique join* is one where given a row from the initial table or tables (called *host tables* on the left of the join panel) matches at most one row in the joined in target table (on the right tab). The join must be unique over the entire set of target table joined columns. Note that no one column must be unique in the target table.

Optimization and avoiding the multiplicative effect

DB2 Web Query allows developers who know that they have a unique join to the target table to configure their report with this knowledge. As a result, DB2 Web Query can potentially reduce the number of SQL statements that are generated, thereby improving performance.

On the Report options tab, the developer can select the Use database optimization check box as shown in Figure 14-3 on page 318. Doing so instructs DB2 Web Query that it is safe to pass the entire report down to the database engine in a single SQL statement. Note that choosing the Use database optimization option, when there is no unique join to the target table, can result in incorrect results in the report (due to the multiplicative effect described earlier). Consequently, it is important to select this option only when you are certain of the uniqueness of the underlying data.

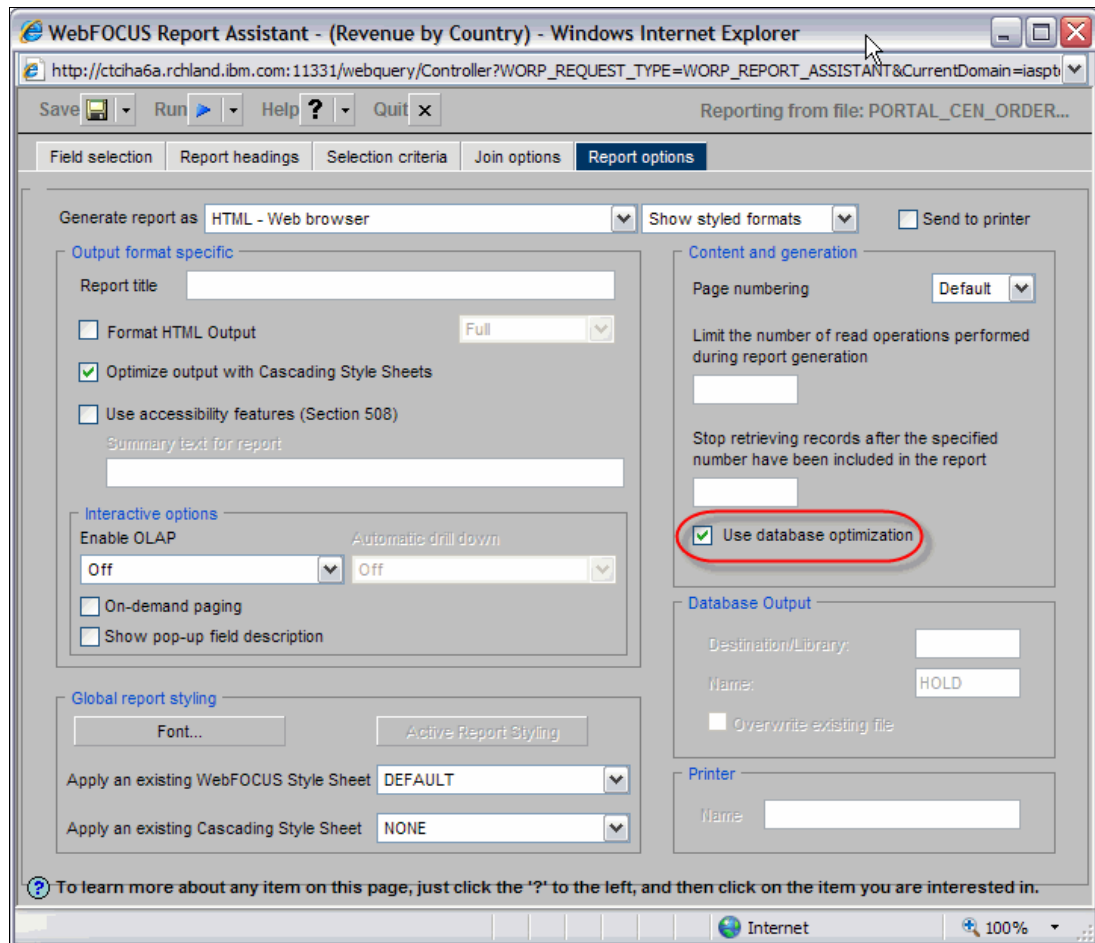


Figure 14-3 Database optimization setting

To determine whether it is safe to select the Use database optimization check box and avoid the multiplicative effect, you can use one of two verification methods:

- Database constraint verification

If either a primary or unique database constraint is in place for a table against a specific column or combination of columns, the database guarantees data uniqueness based on that constraint. In such cases, it is safe to select the Use database optimization check box for that join.

- Manual verification

Tables might have unique data without strict database constraint enforcement. You can also verify that the data is unique by inspection or knowledge of the data. The database administrator can run queries similar to the following example to see if the current state of the data is unique:

```
SELECT
CASE WHEN count(*) = count(distinct JoinColumn1||JoinColumn2||...||JoinColumnN)
then 'Unique' else 'Not Unique' END
FROM target_table
```

If the state of the data is currently unique, the administrator must use their judgement and knowledge of the data to decide if future states of the data are also unique. If you are not sure if future states of the data will be unique over the join columns, then it is better to err on the side of caution and not select the Use database optimization check box.

Sorting optimization

If any of the conditions listed in Table 14-6 are true, DB2 Web Query performs the sorting.

Table 14-6 Sorting and aggregating disablers

Sort disablers	Description
DB2 Web Query managed join	See Table 14-5 on page 317
Sort on a non-translatable DEFINE field	Any request that contains a sort on a non-translatable DEFINE field
Interface-managed join	SQL sorts the answer set by the table's primary key, causing DB2 Web Query to re-sort for the display report.

Aggregation optimization

If any of the conditions listed in Table 14-7 are true, DB2 Web Query performs the aggregation.

Table 14-7 Aggregation disablers

Aggregation disablers	Description
DB2 Web Query managed join or sort	See Table 14-5 on page 317 and Table 14-6.
DB2 Web Query managed row selection	Some WHERE clauses are not passed to the database engine.
Non-direct SQL operators	The request contains direct operators, such as PCT., TOT. that cannot be translated into SQL.
Aggregation on a non-translatable DEFINE	Aggregating on a DEFINE field that cannot directly translate to SQL.

Other ways to influence optimization

The conditions that we discuss in the previous sections help dictate whether DB2 Web Query or the database engine performs the bulk of the report processing. In addition, there are techniques that you can use to influence this behavior and improve report response time.

Creating SQL views

During DB2 Web Query report development, you might encounter situations in which a report does not fully translate to an SQL statement. If this happens, one technique to help move toward full DB2 Web Query optimization is to create an SQL view with all of the appropriate selection, join, and aggregation syntax specified. Next, create the synonym for the view, and in your report definition, select that view, rather than the table or tables from the database description list. This instructs DB2 Web Query to use the view and push the selection, join syntax, and aggregation to the database engine. This particular technique is used to tune a problem report in 14.5, "Performance case study" on page 327.

The advantages of using views are many. The following list gives an idea of what SQL views can do:

- ▶ Provide the ability to specify additional join types that are not supported by the DB2 Web Query product

DB2 Web Query allows you to specify inner joins and left outer joins. With SQL views, you can define those and the following join types:

- Right Outer
- Left Exception
- Right Exception

- ▶ Allow you specify CASE statements to handle more complex, conditional logic
- ▶ Provide fullselect support

An SQL *fullselect* is the term for generating an SQL result set by combining multiple SELECT statements using the UNION, INTERSECT, or EXCEPT operators. This is a feature that helps you solve more complex business requirements.

- ▶ Provide Common Table Expression (CTE) and recursive SQL support

CTEs can be thought of as temporary views that exist only during the execution of an SQL statement. When defined, the CTE can be referenced multiple times in the same view. This can be used to reduce the complexity of the view, making it easier to comprehend and maintain.

Among the V5R4 enhancements for DB2 for i5/OS was the ability for a CTE to reference itself. This feature provides the mechanism for recursive SQL, which is especially useful when querying data that is hierarchical in nature, such as bill of materials, organizational charts, and airline flight schedules.

For more information about recursive CTEs, refer to the article “V5R4 SQL Packs a Punch,” which you can download from the Web at:

http://www-03.ibm.com/servers/eserver/series/db2/pdf/rcte_olap.pdf

- ▶ Allow business logic to reside in the database layer, making it available to all users and all SQL interfaces
- ▶ Provide an database abstraction layer
- ▶ Provide security granularity

When the views are in place, you can restrict users from accessing tables directly and only allow access through the views. Because both selection and projection can be specified on the views, you can control specific rows and columns that users (or groups of users) can access.

Important: Views should not be confused with indexes. They are implemented as non-keyed logical files. This means that views, unlike indexes, have no access paths and thus, no access path maintenance. If you or your database administrator (DBA) are concerned about access path maintenance, this is not an issue with views.

Establishing and using foreign key relationships

If you have foreign key relationships explicitly defined in i5/OS database, you can take advantage of a DB2 Web Query feature to include with the selected table's synonym, every table related to the selected table by a foreign key. The resulting multi-table synonym describes all of this table's foreign key relationships. This can greatly simplify the process of creating reports with joins. In fact, in this scenario, you do not even define a join in your report. All columns from the related tables are displayed on the “Field Selection” tab and you simply select the columns that you want to include in the report. Through the foreign key

relationships, DB2 Web Query is able to understand the join syntax that needs to be created and generates the appropriate SQL statement.

The process of creating synonyms with foreign key relationships is discussed in “Benefits of referential integrity for metadata” on page 38.

Creating materialized query tables

Materialized query table (MQTs) provide another method of improving performance of your queries. An MQT is a table that contains the results of a previously run query, along with the query’s definition. It provides a mechanism for improving the response time of complex SQL queries. While it can be thought of as a summary table, what sets an MQT apart from a regular summary table is the fact that the SQE optimizer is aware of it and its relationship to the query and base tables that were specified when it was created and populated. This means that the optimizer considers using the MQT in the access plan of subsequent similar queries if it determines that it is appropriate to do so. Because the MQT is already created, populated, joined, aggregated, and sorted, this can result in significant performance improvements for complex queries.

It is important to understand that an MQT is a table (a physical file with an object type of *FILE) that resides in a library (schema) in the System i environment. Because of this, it can be accessed directly, just like any other table on the system. While it is usually the optimizer’s job to select and implement an MQT in the access plan of a query request, DB2 Web Query has the ability to create metadata against an MQT. This means that you can create DB2 Web Query reports that access an MQT directly, and not be forced to rely on the optimizer to select the MQT.

To obtain this behavior, you must first create the metadata for the MQT:

1. From the DB2 Web Query home page, right-click your reports folder and select **Metadata**.
2. In the Data Adapter browser window, from the left navigation pane, expand the **DB2 cli** folder and click ***LOCAL**. Select **Create Synonym**.
3. In the Select Synonym Candidates for DB2 CLI (*LOCAL) pane (Figure 14-4), specify the collection and select the **MQTs** check box. Click **Next**.

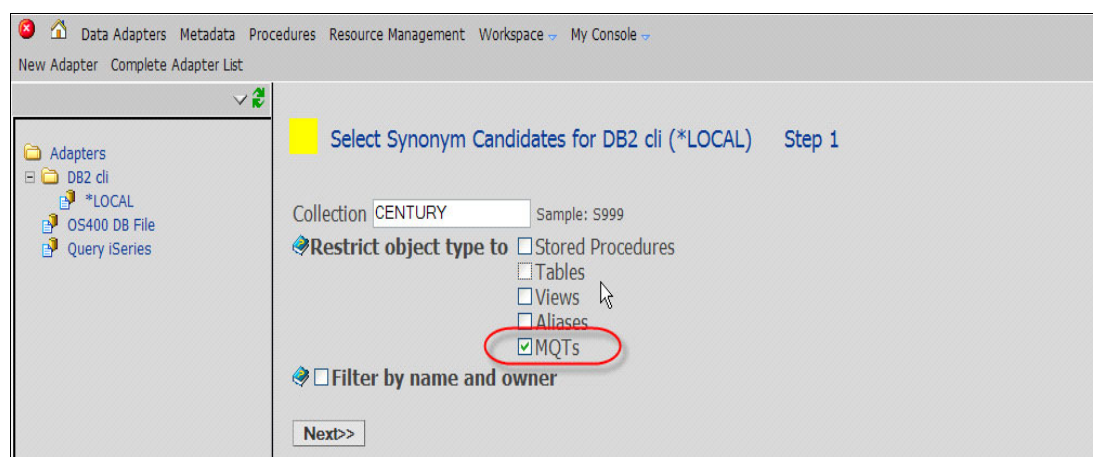


Figure 14-4 Creating a synonym for MQT - Specifying the collection

4. In the Create Synonym for DB2 cli pane (Figure 14-5), specify a prefix and suffix, select the MQTs that are listed, and click the **Create synonym** button.

Create Synonym for DB2 cli (*LOCAL) Step 2 of 2

☐ Cardinality ☐ With foreign keys ☐ Dynamic columns ☐ One-part name

Select application: baseapp Prefix: century_ Suffix: _MQT

☐ Customize data type mappings

☐ Overwrite existing synonyms **Create synonym**

Default synonym name	Owner/Schema	Table name	Type
COUNTRY_STATE_MEASURE_MQT	CENTURY	COUNTRY_STATE_MEASURE_MQT	MQT

Figure 14-5 Creating synonym for MQT - select MQT

After the synonyms are created for the MQTs, you can create reports against them as you would for any other table object on the system.

While their potential performance efficiencies are appealing for DBAs, query developers, and report users, MQTs have several attributes and limitations that must be understood prior to implementation. Probably most important, you must know that MQTs are not automatically maintained. This means that as the base tables used to populate the MQT change, the data in the MQT does not also change. If the data in your reports must be up to date, the MQT must first be manually refreshed. Therefore, before implementing MQTs, you must be willing to accept some level of data latency. This is can be acceptable for queries that report on data that is historical in nature. However, if your report requires data that is dynamic and up-to-the-minute, you might come to the conclusion that an MQT is not the right fit.

For more information about MQTs, see the white paper *Creating and using materialized query tables (MQT) in IBM DB2 for i5/OS*, which you can find on the Web at:

<http://www-304.ibm.com/jct09002c/partnerworld/wps/servlet/ContentHandler/SROY-6UZ5E6>

14.2.3 Remote database access considerations (including cross-system joining)

One feature of the DB2 Web Query base product is the ability to access data on remote DB2 for i5/OS systems or partitions. This feature, which is available only for the DB2 CLI adapter, gives you the option to pull data for your reports in one of the following ways:

- ▶ Local: All the data resides on the local system
- ▶ Remote: All the data resides on one remote system or partition
- ▶ Cross system: Data is spread across the local system or one or more remote systems or partitions.

If full DB2 Web Query optimization can be performed for Local and Remote database access, and no other external factors are negatively influencing query performance, you can expect your reports to perform reasonably well. Again, this is because DB2 Web Query can generate one SQL statement and allow the DB2 for i5/OS database engine to process the request.

However, you must be aware of the performance implications if the data is spread across two or more System i machines or partitions. In this scenario, DB2 Web Query generates a separate SQL statement for each connection and submits the statement to each connection.

The answer set from each of these multiple sources is then joined together by the Reporting Server. Depending on how the joins are specified, this can be a lengthy and time consuming process. Consider the following scenario.

A fact table (ORDERS) has 32,283 rows and is joined to a dimension table (STORES) that has 116 rows. If both of the tables are on the local (or even remote) system, DB2 Web Query generates a single SQL statement and submits it to DB2 for i5/OS. Example 14-1 shows how the statement looks.

Example 14-1 Single SQL statement

```
SELECT T2."Country",T2."State", SUM(T1."LineTotal")
FROM CENTURY/ORDERS T1,CENTURY/STORES T2
WHERE (T2."StoreCode" = T1."StoreCode")
GROUP BY T2."Country",T2."State"
ORDER BY T2."Country",T2."State"
FOR FETCH ONLY
```

Since one statement is generated and the database engine handles all the joining, grouping, and ordering, you can expect this report to run quite efficiently.

However, if the dimension table STORES resides on another system and Report Assistant is used to create a nearly identical report (the only difference being a cross system join specification to STORES), DB2 Web Query now creates two SQL statements, one for each system.

► Local system

Example 14-2 Local system SQL statement

```
SELECT T1."StoreCode",T1."LineTotal"
FROM CENTURY/ORDERS T1
FOR FETCH ONLY
```

► Remote system

Example 14-3 Remote system SQL statement

```
SELECT T5."Country",T5."State"
FROM "CENTURY"/"STORES" T5
WHERE (T5."StoreCode" = ?)
FOR FETCH ONLY
```

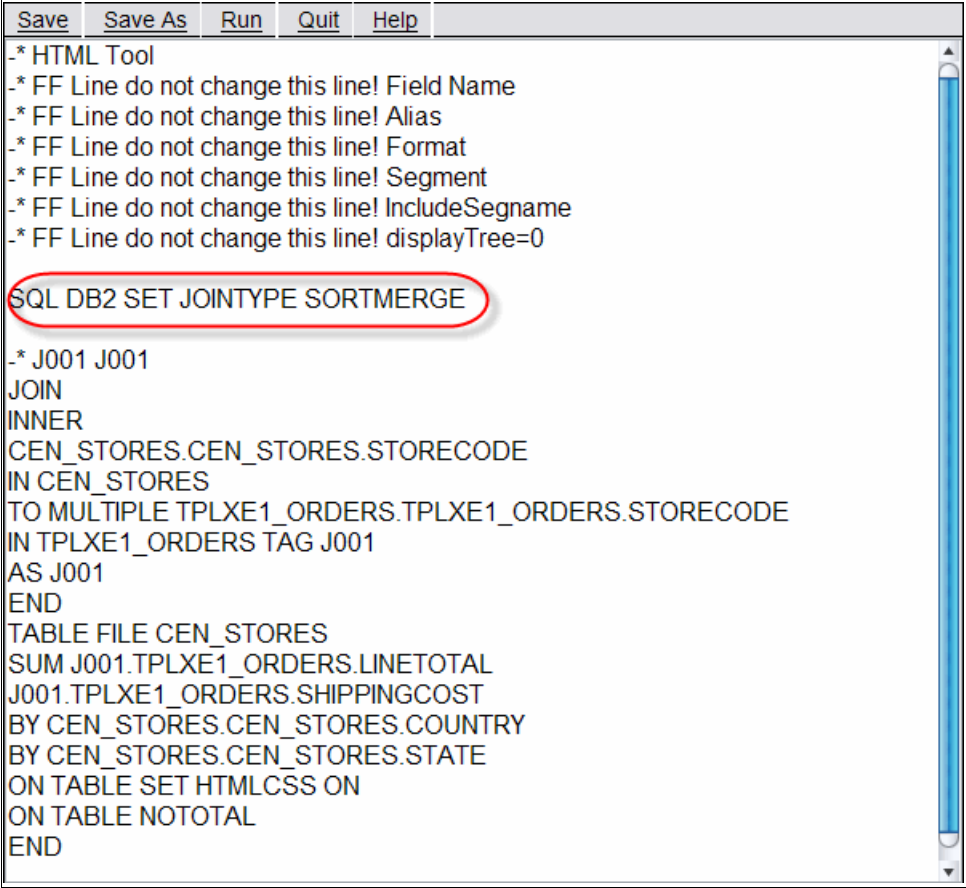
Inefficiencies occur because the remote SQL statement is submitted multiple times on the remote system. The local system fetches each row from the (local) ORDERS table result set and generates the appropriate local selection to return the matching row from the remote STORES table. With this implementation, the report takes significantly longer before the results are displayed to the browser.

If the report is modified to specify STORES (with only 116 rows) as the base table, the join fan-out is greatly reduced. This results in a report that runs significantly faster, but is still not nearly as fast as when all tables are on same system. Therefore, we generally recommend that you avoid creating reports with cross-system joins when the tables have a substantial number of rows. However, if such an implementation is required, consider the following suggestions to enhance the report's performance:

- ▶ Eliminate the report's requirement of multiple connections by making local copies of the remote tables. After all tables are on one system, join optimization can be enabled and DB2 Web Query attempts to generate a single SQL statement.
- ▶ Determine which of the tables has the smallest number of rows (after applying local selection) and make that the base table of the report. This reduces the fan-out effect of the join, thereby reducing the number of rows that must be retrieved from the other (larger) tables.
- ▶ Edit the report source and add the following line:

SQL DB2 SET JOINTYPE SORTMERGE

Figure 14-6 shows an example of this setting specified in the report source. This setting impacts the number of FOR FETCH lines when one of the tables is remote. In effect, it turns all of the tables into local internal tables and sort merges them.



```

Save Save As Run Quit Help
-* HTML Tool
-* FF Line do not change this line! Field Name
-* FF Line do not change this line! Alias
-* FF Line do not change this line! Format
-* FF Line do not change this line! Segment
-* FF Line do not change this line! IncludeSegname
-* FF Line do not change this line! displayTree=0
SQL DB2 SET JOINTYPE SORTMERGE
-* J001 J001
JOIN
INNER
CEN_STORES.CEN_STORES.STORECODE
IN CEN_STORES
TO MULTIPLE TPLXE1_ORDERS.TPLXE1_ORDERS.STORECODE
IN TPLXE1_ORDERS TAG J001
AS J001
END
TABLE FILE CEN_STORES
SUM J001.TPLXE1_ORDERS.LINETOTAL
J001.TPLXE1_ORDERS.SHIPPINGCOST
BY CEN_STORES.CEN_STORES.COUNTRY
BY CEN_STORES.CEN_STORES.STATE
ON TABLE SET HTMLCSS ON
ON TABLE NOTOTAL
END

```

Figure 14-6 SORTMERGE setting

14.3 Query/400 adapter performance

Another feature of DB2 Web Query is the ability to run most existing i5/OS Query/400 queries. This is done by using the Query/400 adapter that is provided with the base offering of 5733-QU2. This adapter functions by running the Query/400 query during both the synonym creation process and report execution. This means that if you have an existing Query/400 object that does not perform well and takes several minutes to run to completion, it requires just as much time (and perhaps more) to create the DB2 Web Query synonym. After the

synonym is created and you run the new DB2 Web Query report, you can again expect it to require at least the same amount of time to run.

Because the adapter is running the existing query, you are limited in the tuning that can be done to make each of the DB2 Web Query processes perform better. In most cases, the sole tuning knob ensures that the appropriate indexes are in place for the optimizer to use when creating the access plan for the query. Refer to 14.4.2, “Indexes” on page 326, to learn about approaches for implementing an indexing strategy.

As we mentioned earlier in this chapter, Classic Query Engine (CQE) processes non-SQL database requests such as those from Query/400 queries. CQE provides index advisory information for query selection. Missing from CQE is the ability to create index advisories for joining, grouping, and ordering. Analysis and tuning, which are more manual in nature, are required for these types of requests. If your Query/400 reports are not performing to your satisfaction, make sure that the indexes exist over all join fields, order by fields, and fields that are aggregated.

14.4 DB2 for i5/OS optimization

By now you should be convinced that getting DB2 Web Query to hand off as much of the processing as possible to the database engine is a key factor in achieving optimal report performance. There are also factors within the i5/OS database itself that can affect efficiency.

These types of factors are usually the responsibility of the DBA and not the users; however, a little knowledge of these factors and how they can affect efficiency enables a user to communicate effectively with the DBA.

When the users have done everything that they possibly can in the efficiency spectrum to ensure that the report definitions are as good as they can be, these are the factors that can still influence the overall efficiency of the request. The following DB2 for i5/OS factors can affect efficiency:

- ▶ Database design
- ▶ Indexes
- ▶ Available hardware
- ▶ Number of concurrent users

14.4.1 Database design

Good performance starts with a good database design. If your database structure is flat in nature and not normalized, the files are likely to have redundant data that is difficult to navigate, retrieve, and maintain. The complexities of retrieval and maintenance are often handled by high-level language (HLL) program written in RPG or COBOL. This approach works when such programs are the only interface to the data and can hide the complexity from users. However, problems and confusion are likely to result when you expose your database and allow other interfaces, such as DB2 Web Query, to access and report against the data. In the following list, we highlight the shortcomings of a poorly designed database:

- ▶ Complexity

The application programs contain most, if not all, business rules and data relationships. Therefore, the data is difficult to query because it is hard to understand rules and relationships. Because of the complexity, programmers (and not users) are forced to develop the reports. Often the reports themselves are HLL programs that can be complex

and difficult to maintain. Programmers can find themselves constantly creating new copies of reports to satisfy a seemingly endless list ad hoc reporting requirements by the user.

- Lacks flexibility

The database is difficult to maintain, adapt, and expand. Repeating groups must be squeezed into a single record, which places limits on the number of repeating groups that can fit into a single record format.

- Less optimal access methods

Performance of these reports can suffer because of the database complexity. Programmatic selection, joining, ordering, and grouping are often implemented with a row-at-a-time coding techniques. Record blocking and set-at-a-time processing are not used, and therefore, the report performance is not maximized.

If this describes your current database design, you might want to consider investing in a robust data modeling tool that can help you implement a database design that is both functional and efficient.

14.4.2 Indexes

Like any other application that accesses information from the database, efficient DB2 Web Query performance heavily depends on having the right indexes in place. When your database is in production, it might be difficult to implement recommended database design practices such as database normalization. However, at this point, you can still implement an indexing strategy that helps to optimize the performance of your query reports.

Indexes over your database tables have the following advantages:

- Provide statistics to the optimizer

Indexes against the queried tables give the optimizer a better opportunity to select the most efficient access method. This is because they provide relevant statistics and information, such as the average number of duplicate values and column cardinality of the tables being queried. Such useful information provided to the optimizer results in a better access plan and a better performing query.

- Improve efficiencies

The optimizer can choose to use the index during implementation, thus avoiding more costly implementation alternatives such as table scans or creation of temporary structures. If a table scan is performed, every row in the database table must be read. Depending on the size of the tables and the complexity of the query, this can be a lengthy process and can consume a significant share of system resources.

- Ensure uniqueness

A unique index on a column ensures that no duplicate values exist for that column.

Using indexes might result in the following ramifications:

- As mentioned previously, indexes can speed data retrieval. In some cases, requests might only need to use the index, never accessing the actual data (a condition referred to as *Index Only Access*).
- Even secondary indexes can be used in the case of data selection statements.
- Indexes add overhead to a database. Indexes must be maintained by the database whenever the data in the underlying table changes. Sometimes DBAs are reluctant to add indexes for reporting application efficiency if the database is one that is not dedicated to reporting purposes.

Indexing strategies is a broad topic that is covered in detail in the white paper *Indexing and Statistics Strategies for DB2 for i5/OS*. You can download this paper from the Web at the following address:

<http://www.ibm.com/servers/enable/site/education/ibo/record.html?indxng>

In addition, the following tips can help you get started:

- ▶ Take a proactive approach and make sure there are indexes available over all of the selection, joining, ordering, and grouping columns of your queries.
- ▶ In a reactive mode, run your queries and use the available database feedback mechanisms to determine what indexes the optimizer wants created.

When a query is executed, the database engine provides index advice during the optimization phase. This occurs when it determines that a useful index does not exist against the table or tables that are being queried. It makes a recommendation that the index be created. Afterward, this advice can be obtained from various sources, including the following sources:

- Index Advisor
- SQE Plan Cache
- SQL Plan Cache snapshot, 3020 record
- SQL Performance Monitor, 3020 record
- Visual Explain

The index advisories can be extracted from these sources of optimizer feedback and used to create the recommended indexes.

Important: When an advised index is created and the query is run again, you might observe that the optimizer does not use that index during implementation. Indexes are sometimes advised for the information they can provide to the optimizer. This information is used to help the optimizer cost each access method. It might also help the optimizer determine that the advised index is not the optimal one to use during query execution.

For more information about SQL and database performance analysis, refer to *OnDemand SQL Performance Analysis Simplified on DB2 for i5/OS in V5R4*, SG24-7326.

14.5 Performance case study

In this section, a case study is conducted against a long running DB2 Web Query report to delve further into performance analysis and demonstrate available optimization techniques. The objective is to determine why the particular report has an excessive runtime and provide example approaches in order to reduce the overall runtime.

In this section, we describe the following steps:

1. Identify a long running report.
2. Perform analysis, looking for optimization disablers.
3. Determine report tuning options.
4. Create an SQL view and synonym, and change the report to use view.
5. Create a new report based on the SQL view.
6. Perform additional database analysis and tuning.

14.5.1 Identifying a long running report

During unit testing, a report that was created using Report Assistant and the SQL CLI adapter was identified as one that required an excessive amount of time to run to completion. This report, named “Order Revenue within 180 days by Year, Month”, took several minutes to run. However reports that were similar in nature, meaning they had the same general format and approximately the same number of rows returned, ran in seconds.

14.5.2 Performing analysis and looking for optimization disablers

To help determine the reasons behind the substantial run time, DB2 Web Query provides the Run w/SQL Trace option (Figure 14-7) to run the report and generate an SQL trace. The trace statements that are displayed help reveal any optimization disablers that might be present in the report.



Figure 14-7 Run w/SQL trace

When the report is run using the Run w/SQL Trace option, the SQL trace information is displayed as HTML output to the browser as shown in Figure 14-8 on page 329.

This trace reveals a couple of interesting points. First, it shows that part of the report cannot be translated to SQL. In this case, translation failed for the Defined field DAYS_DIFF because of an unsuccessful IF/WHERE Test. Second, notice how the trace shows the SQL statement that is generated (and what is ultimately submitted to the database engine). Even though part of the translation failed, the Reporting Server was able to construct a meaningful SQL statement.

No HTML Output!

```

13.12.17 BR (FOC2598) FOCUS IF/WHERE TEST CANNOT BE PASSED TO SQL : DAYS_DIFF
13.12.17 BR (FOC2590) AGGREGATION NOT DONE FOR THE FOLLOWING REASON:
13.12.17 BR (FOC2596) ONE OR MORE EXPRESSION(S) CAN NOT BE TRANSLATED TO SQL
13.12.17 AE SELECT T1."SHIPDATE",T1."CUSTKEY",T1."REVENUE_WO_TAX",
13.12.17 AE T1."YEAR",T1."MONTH",T2."CUSTOMER" FROM "STAR1G"/"ITEM_FACT" T1,
13.12.17 AE "STAR1G"/"CUST_DIM" T2 WHERE (T2."CUSTKEY" = T1."CUSTKEY") AND
13.12.17 AE (T1."SHIPDATE" < '1998-01-31') AND (T2."CUSTOMER" BETWEEN
13.12.17 AE 'Customer#000100000' AND 'Customer#000200000') ORDER BY
13.12.17 AE T1."YEAR",T1."MONTH" FOR FETCH ONLY;
...RETRIEVAL KILLED
0 NUMBER OF RECORDS IN TABLE= 0 LINES= 0

```

Not translated to SQL (and why)

Translated SQL statement

Figure 14-8 Report trace results

Using the trace, you can conclude that the problem with this report is the use of the define field DAYS_DIFF. The line in the trace “FOCUS IF/WHERE TEST CANNOT BE PASSED TO SQL. DAYS_DIFF” indicates that DB2 Web Query attempted (and failed) to generate an equivalent search condition for the SQL statement’s WHERE clause (Figure 14-9).

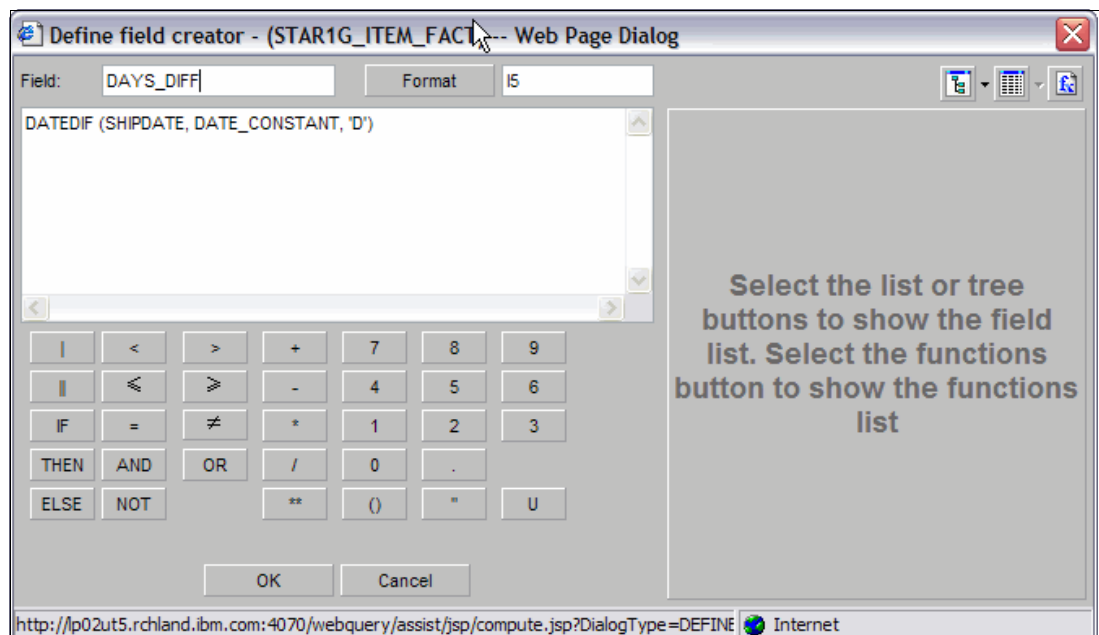


Figure 14-9 Define field DAYS_DIFF

Anytime a define field can be passed to the database engine, a more efficient report results, particularly when the define field is part of the selection process, which it is in the case of our example. If the define field cannot be passed to the database engine, then all the rows from the result set of the generated SQL statement are returned to the Reporting Server, which now is responsible for the selection (and sorting if the report is sorted by the define field). This kind of behavior that can negatively impact performance and should be avoided if at all possible. Again, it must be emphasized that the report is more efficient if the entire database access portion of the report definition can be pushed down to the database engine for processing.

14.5.3 Determining report tuning options

Tuning this particular report basically means selecting one of the following actions:

- ▶ Eliminate the culprit define field.
- ▶ Modify the report in an attempt to enable SQL translation for the define field.
- ▶ Push the define field logic down to database using SQL views.

For this example, the define field is absolutely required to deliver the necessary information in the report (and therefore cannot be removed). In addition, repeated efforts to fix the translation for the define field prove to be futile. Therefore, only the third option remains to push the define field logic down to a database using SQL views.

14.5.4 Creating an SQL view and synonym

The next step is to create an SQL view that contains all the required tables, join syntax, columns, and selection, including the DAYS_DIFF field selection that DB2 Web Query was unable to directly translate to SQL. To provide the ability to calculate the number of days between the date 1998-01-31 and the value of the SHIPDATE column, a new result column is added to the view. Example 14-4 shows this result column, also named DAYS_DIFF.

Example 14-4 DAYS_DIFF result column

```
DAYS('1998-01-31') - DAYS(T1.SHIPDATE) AS DAYS_DIFF
```

To create the new view:

1. From iSeries Navigator, open a connection to the System i environment and open a Run SQL Script window.
2. Type the CREATE VIEW statement or use the SQL selection statement that is displayed in the trace as a base for this view definition and make the necessary modifications. Copy and paste the statement into the Run SQL script window and begin making the changes. For the copy and paste method, Figure 14-10 on page 331 shows more details about what to add and remove from the selection statement to form the CREATE VIEW statement.

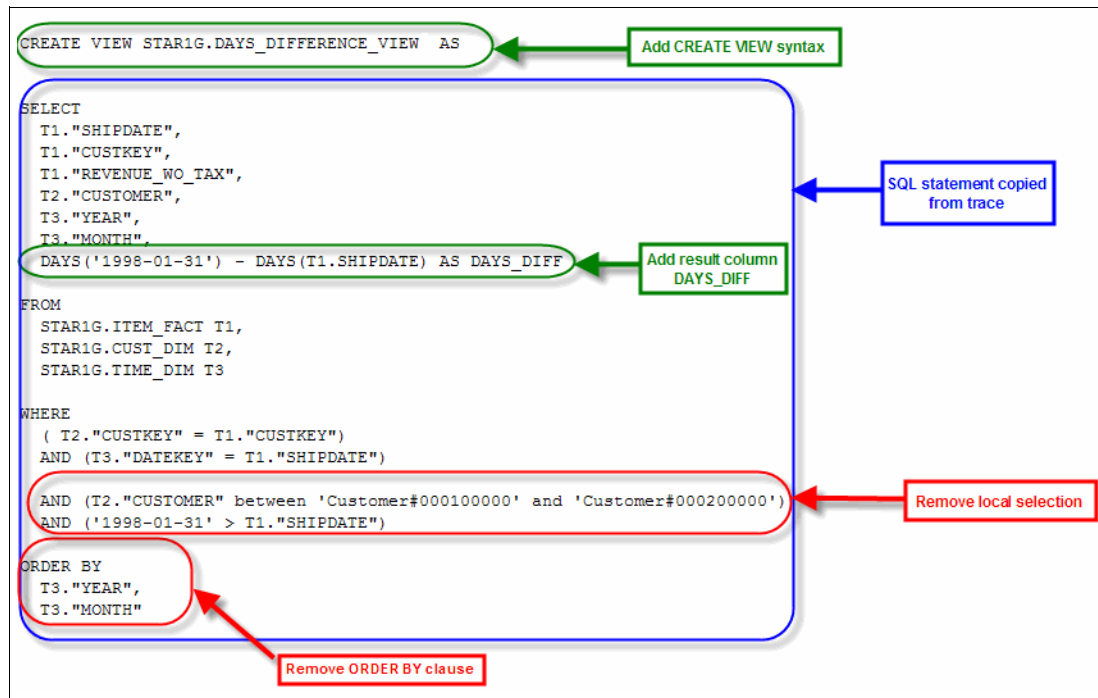


Figure 14-10 Anatomy of the CREATE VIEW statement

Notice that the local selection is removed from the CREATE VIEW statement. This is added to the statement that references the view. Also observe that the ORDER BY clause is removed. This is because views cannot be ordered. Ordering is specified in the report definition.

After all the necessary modifications are made, Example 14-5 shows how the CREATE VIEW statement should look.

Example 14-5 Create SQL view

```
CREATE VIEW STAR1G.DAYS_DIFFERENCE_VIEW
AS
  SELECT T1."SHIPDATE",T1."CUSTKEY",T1."REVENUE_WO_TAX",
        T2."CUSTOMER",T3."YEAR",T3."MONTH",
        DAYS('1998-01-31') - DAYS(T1.SHIPDATE) as DAYS_DIFF
  FROM STAR1G.ITEM_FACT T1, STAR1G.CUST_DIM T2,STAR1G.TIME_DIM T3
  WHERE (T2."CUSTKEY" = T1."CUSTKEY") AND (T3."DATEKEY" = T1."SHIPDATE")
```

3. Execute the statement to create the view.
4. From the browser, open the DB2 Web Query metadata window and create a synonym against the new view.

14.5.5 Creating a new report based on the SQL view

When the new view and its synonym are in place, a new report is created to access the contents of the view:

1. From DB2 Web Query home page, select **Report Assistant** to create a brand new report.
2. In the Select from available database descriptions window (Figure 14-11), select the new view. Click **OK**.

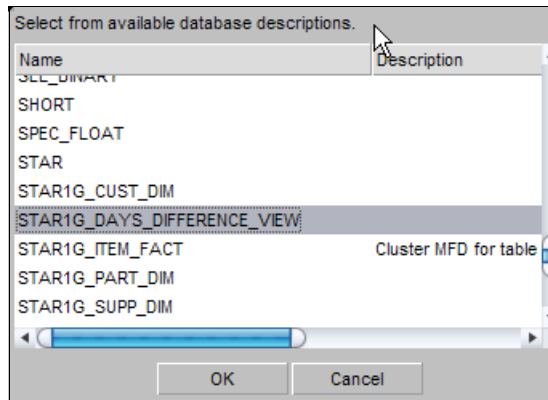


Figure 14-11 Report Assistant - Selecting a new view

3. Continue creating the new report with the view, specifying the same field format and selection criteria fields (as the original report). Examples are shown in Figure 14-12 and Figure 14-13 on page 333.

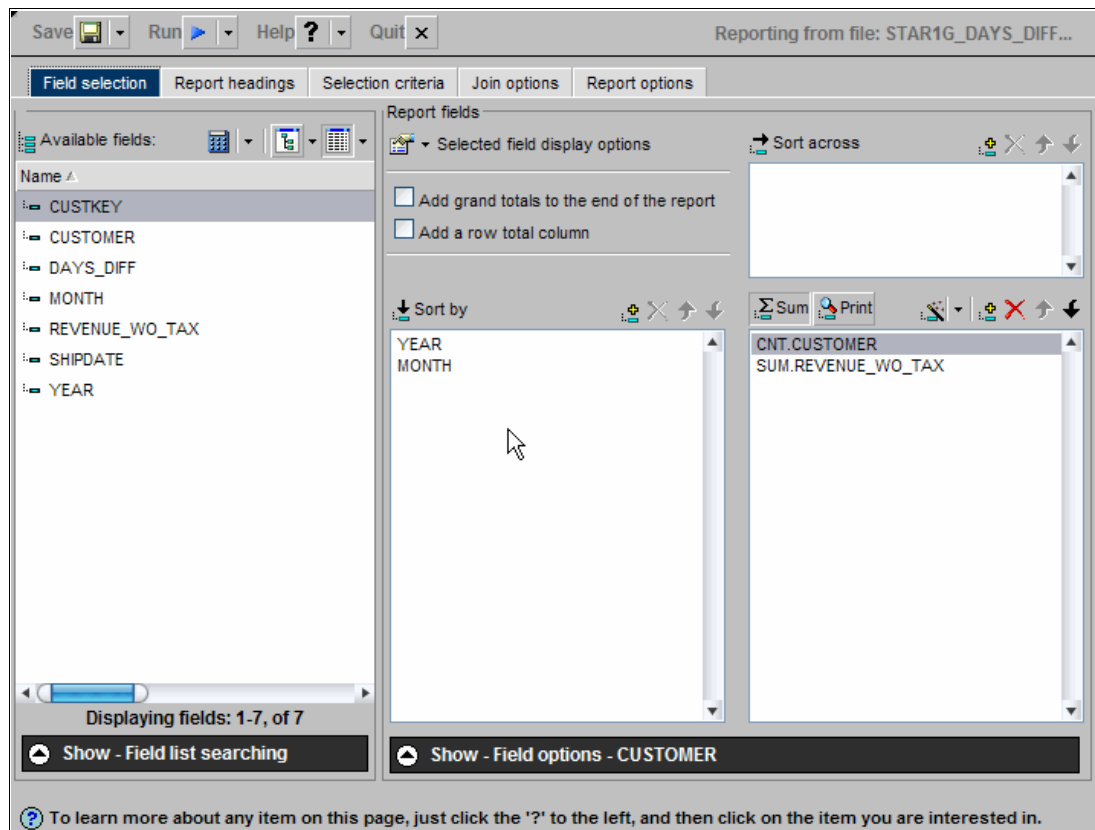


Figure 14-12 Report using SQL view - Field selection tab

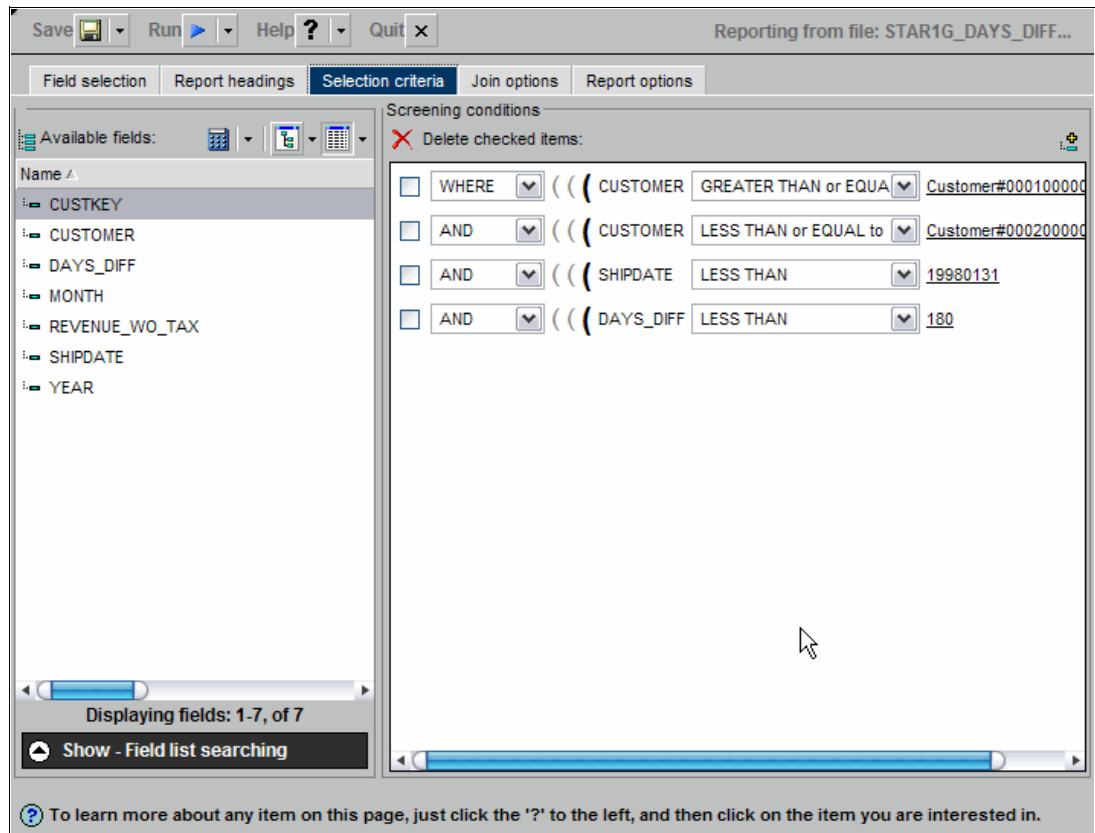


Figure 14-13 Report using SQL view - Selection criteria tab

4. Save the report and click **Quit**.
5. Run the new report.
6. Record performance measurements.

In the case of this example, significant improvements in performance are observed. In fact, simple benchmark measurements reveal that the report, using the view, runs over 10 times faster than the original version of the report. This is not meant to imply that you will experience the same improvements if you conduct such a tuning exercise. It is only intended to serve as an example of the kinds of efficiencies that you can obtain. Many factors can affect report performance, so your results might vary.

14.5.6 Performing additional database analysis and tuning

When your report is using the SQL view and all database processing is being handled by the database engine, your work might not yet be done. You can potentially gain further efficiencies by performing regular database analysis and tuning. You can use the following tools and technologies to assist in this effort:

- **SQE Plan Cache**

First made available in V5R2, the SQE Plan Cache is an internal, matrix-like repository that is used to store all of the statements and access plans implemented by SQE. In V5R4, an interface to this information has been made available through the iSeries Navigator toolset. From this interface, you can find the SQL statement generated by DB2 Web Query and begin performing your analysis.

► Visual Explain

Visual Explain provides a graphical representation of the optimizer implementation of a query request. The query request is broken into individual components with icons that represent each unique component. Visual Explain also includes information about the database objects that are considered and chosen by the query optimizer. Visual Explain's detailed representation of the query implementation makes it easier to understand where the greatest cost is incurred.

► Index Advisor

Introduced in V5R4, this feature provides an easy and quick interface to index advisories that are issued by the optimizer. If the optimizer determines that a permanent index against a reference table might be beneficial, it returns the key columns necessary to create the suggested index. The data of the system-wide Index Advisor is placed into the SYSIXADV table in the QSYS2 schema.

► Database monitor

The Database Performance Monitor is a set of integrated tools that is used to collect database-specific performance information for all SQL requests. It can be thought of as an SQL tracing facility, one that tracks all SQL statements, access plans used to implement the statements, resources used, and subsequent performance results. All this information is stored in a database tables, where it can be analyzed and used to identify and tune performance problem areas.

► Materialized query tables

As mentioned previously, an MQT is a DB2 table that contains the results of a query, along with the query's definition. Because the selection, joining, and aggregation have already been performed and the results stored in the MQT, great efficiencies can be gained if the optimizer uses this table for implementation.

► Index Only Access

The database optimizer can use Index Only Access if all of the columns specified in the SQL statement are represented in the index as key columns. Because all of the columns that are necessary to satisfy the request are present in the index, the database engine does not have to perform random access to the table to retrieve this data. The elimination of this additional I/O operation can result in significant improvements in query response times.

Database performance analysis and tuning are rather broad topics that are briefly discussed in 14.4, "DB2 for i5/OS optimization" on page 325. For a more extensive discussion about this subject, refer to *OnDemand SQL Performance Analysis Simplified on DB2 for i5/OS in V5R4*, SG24-7326.

For this exercise, feedback from the optimizer (obtained by locating the statement in the SQE plan cache and launching Visual Explain) suggests the creation of the indexes shown in Example 14-6.

Example 14-6 Indexes created for the case study

```
CREATE INDEX STAR1G.CUST_DIM_CUSTKEY_CUSTOMER
  ON STAR1G.CUST_DIM ( CUSTKEY ASC , CUSTOMER ASC )
  PAGESIZE( 64 ) ;

CREATE ENCODED VECTOR INDEX STAR1G.ITEM_FACT_CUSTKEY_EVI
  ON STAR1G.ITEM_FACT ( CUSTKEY ASC )
  WITH 65355 DISTINCT VALUES ;

CREATE INDEX STAR1G.ITEM_FACT_CUSTKEY_SHIPDATE
```

```

ON STAR1G.ITEM_FACT ( CUSTKEY ASC , SHIPDATE ASC )
PAGE SIZE( 64 ) ;

CREATE ENCODED VECTOR INDEX STAR1G.ITEM_FACT_SHIPDATE_EVI
ON STAR1G.ITEM_FACT ( SHIPDATE ASC )
WITH 65355 DISTINCT VALUES ;

CREATE INDEX STAR1G.TIME_DIM_YEAR_MONTH
ON STAR1G.TIME_DIM ( "YEAR" ASC , "MONTH" ASC )
PAGE SIZE( 64 ) ;

```

These indexes are created and the report is run again. Simple benchmark testing reveals modest improvements to the runtime after these indexes are created.

14.6 Performance benchmark

A performance benchmark was conducted to help determine DB2 Web Query runtime performance expectations relative to the equivalent Query/400 objects and SQL statements.

14.6.1 Objectives

The benchmark had the following objectives:

- ▶ Measure the overhead and resource usage of using DB2 Web Query product to execute existing Query/400 reports
- ▶ Measure the overhead and resource usage of running DB2 Web Query reports versus equivalent SQL statements

14.6.2 Scenarios

The benchmark test bucket included six scenarios, each with a set of eight different queries or statements. Each query was run with ten different sets of host variable values to measure variances in selectivity and cardinality. We measured the following scenarios of queries:

- ▶ Query/400 *QRYDFN objects (using the i5/OS RUNQRY command)
- ▶ DB2 Web Query running the *QRYDFN objects
- ▶ Direct SQL statements
- ▶ DB2 Web Query reports using the DB2 CLI adapter
- ▶ Direct SQL statements with all full open and re-optimization
- ▶ DB2 Web Query reports using the DB2 CLI adapter with all full open and re-optimization

For each measurement, we captured the following data:

- ▶ Total execution time for a given number of loops of the executed queries and reports
- ▶ Performance monitor data
- ▶ Task-profile trace, also known as TRPOF

This is a sample-based trace that queries the processor, at user-defined intervals, to gather data on what is currently running on the processor (or processors).

14.6.3 Database and system configuration

For all of the benchmark measurements, a sample database with approximately 1 GB of data was used.

The measurements were done on a Model 515 System i environment with one or two processors, 7.5 GB of memory, and six disk arms. After running sets of measurements on the system, it was determined that one processor was a better match for the amount of memory and number of disk arms on the system. All measurements reported were done with a one processor configuration.

14.6.4 Metrics

The measurement data was summarized into four key metrics:

- ▶ Minimum average response time per query
- ▶ Maximum throughput
- ▶ CPU usage
- ▶ Memory usage

Minimum average response time per query

The metric for minimum average response time shows the average response time per query for a single user. The results of this metric are shown in Figure 14-14. The average response time (sec/query) is computed as the inverse of the measured throughput (queries/sec) for a single user running back-to-back queries with no think time.

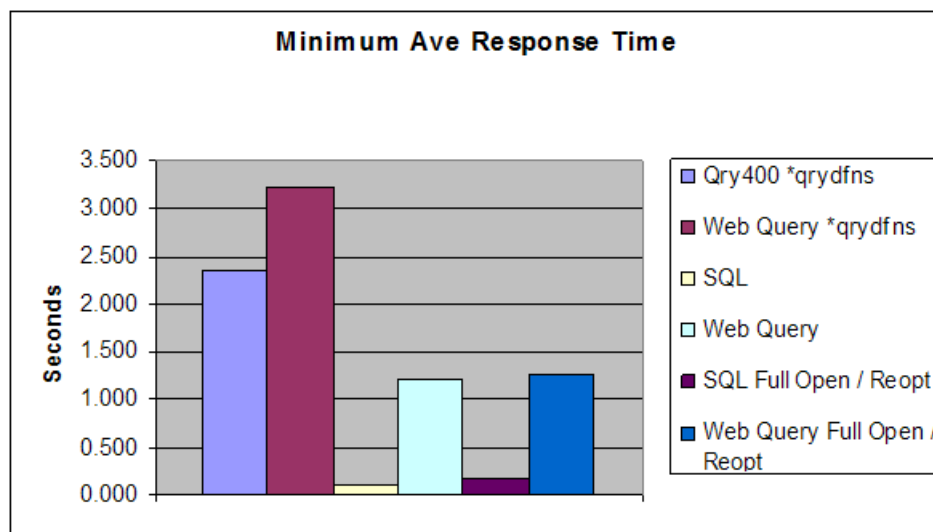


Figure 14-14 Minimum average response time chart

Figure 14-14 shows that there is a moderate response time increase for *QRYDFN objects that are run through DB2 Web Query compared to *QRYDFN run through Query/400. Some response time increase is expected when running from a remote client. As expected, the direct SQL statements have the fastest response time. The response time of the DB2 Web Query reports, although longer than the direct SQL statements, is significantly faster than the response time of the *QRYDFN objects. This is primarily due to the efficiencies of the SQL generated by the DB2 Web Query running through SQE. With all queries incur full open and optimization, more like ad hoc queries, the DB2 Web Query response time is only slightly increased. This is the case due to server jobs being recycled after each DB2 Web Query

report. The extra overhead of more full opens is incurred in the base DB2 Web Query response time numbers as well.

Maximum throughput

The objective of measuring maximum throughput is to show the maximum number of queries per second that ran before reaching a system bottleneck, such as processor or disk utilization. See Figure 14-15. This limit is measured by increasing the number of users who are running queries until no additional throughput is gained.

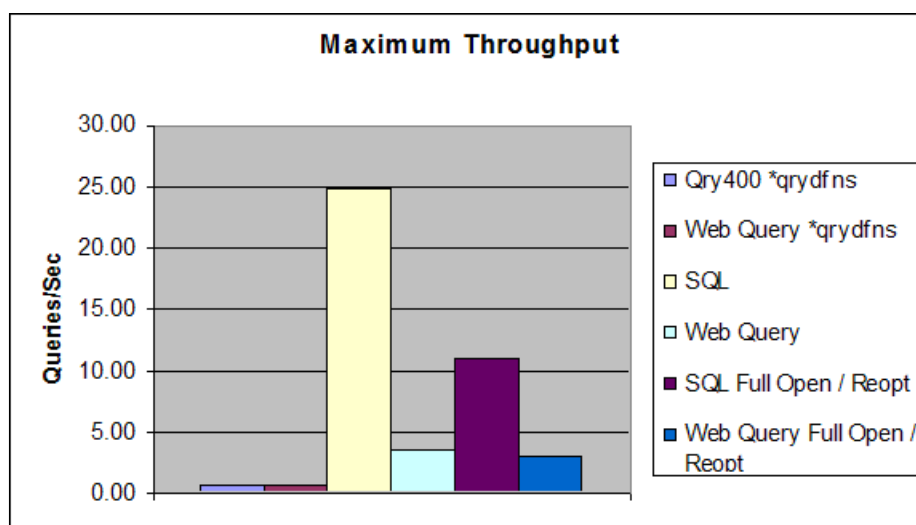


Figure 14-15 Maximum throughput chart

The maximum throughput for *QRYDFN objects that run through the Web Query interface is almost equivalent to the maximum throughput of running the *QRYDFN objects through Query/400. As expected, the executed SQL statements have the greatest throughput. DB2 Web Query introduces additional overhead compared to SQL. However the maximum throughput for the DB2 Web Query reports is much greater than the throughput of the DB2 Web Query *QRYDFN runs. This is due to the efficiencies of the SQL generated by the Web Query product running through SQE. With all queries incurring full open and optimization, more like ad hoc queries, the Web Query maximum throughput is only reduced slightly. This is the case due to server jobs being recycled after each Web Query report. Therefore, the extra overhead of more full opens incurs in the base Web Query throughput numbers as well.

Resource usage: CPU

The chart of CPU resource usage shows the amount of CPU resource used per query on average (Figure 14-16 on page 338). The CPU utilization is measured when the system was running at maximum throughput. The CPU usage is then calculated as the CPU utilization divided by the queries per second at the maximum throughput point.

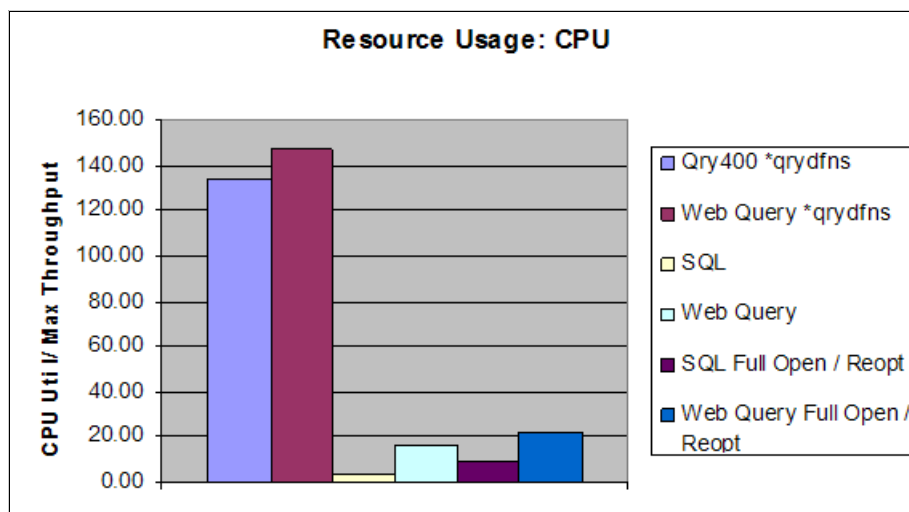


Figure 14-16 CPU usage chart

Figure 14-17 shows a small increase in CPU resource used when *QRYDFN objects are run through the Web Query interface compared to the CPU used by *QRYDFN objects run through Query/400. The chart also shows the dramatic reduction in CPU used by Web Query reports and SQL compared to the *QRYDFN objects. This is due to the efficiencies of SQL running in SQE. The Web Query reports use more CPU than the equivalent plain SQL.

Resource usage: Memory

The chart of memory resource usage shows the minimum amount of base pool memory needed to be able to maintain the corresponding maximum throughput rates shown in the maximum throughput chart (Figure 14-17). The memory requirement is determined by collecting a memory curve. Throughput for a fixed number of users is measured at various base pool sizes. The smallest memory size measured with the high level of throughput is the memory requirement point. DB2 Web Query running *QRYDFN objects compared to *QRYDFN objects run through Query/400 require more memory. This is expected due to the additional server jobs used by DB2 Web Query. As expected, the plain SQL queries require the least amount of memory. DB2 Web Query reports, running at a higher throughput rate, require about the same amount of memory as the *QRYDFN objects run through Query/400.

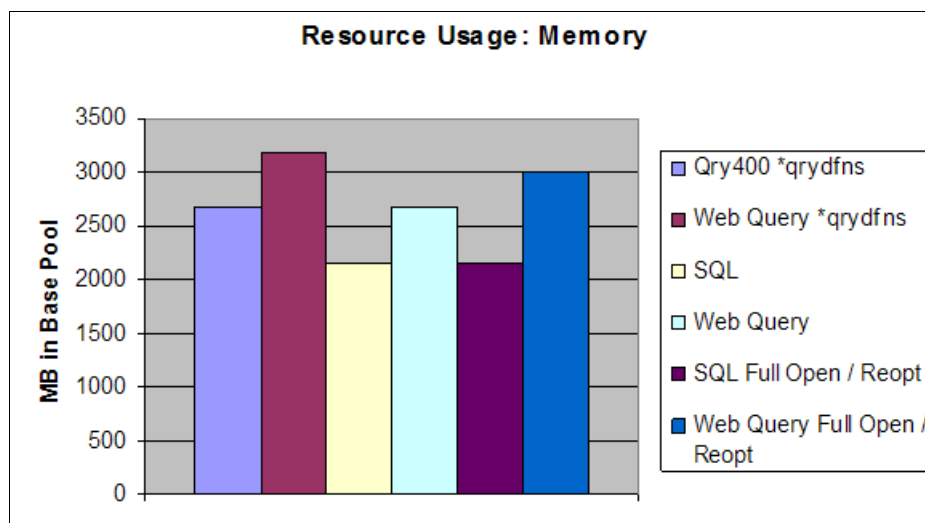


Figure 14-17 Memory usage chart

Figure 14-18 lists the data used in the graphs in this section.

	Minimum Average Response Time (Sec/Query)	Maximum Throughput (Queries/sec)	CPU Util/Throughput	Memory Req'd (MB)
Qry400 *qrydfns	2.346	0.74	134.55	2676
Web Query *qrydfns	3.240	0.64	147.26	3188
SQL	0.116	24.82	3.65	2164
Web Query	1.213	3.60	16.20	2676
SQL Full Open / Reopt	0.189	11.07	8.80	2164
Web Query Full Open / Reopt	1.263	3.11	22.30	3000

Figure 14-18 Benchmark results table

14.6.5 Conclusions

We used the results of the benchmark to derive the following conclusions:

- ▶ Comparisons of the DB2 Web Query *QRYDFN versus the Query/400 *QRYDFN show an increased response time. This result is expected when running to a client application as is the case with DB2 Web Query. The server CPU used is 9% more, and more memory is required.
- ▶ Comparisons of DB2 Web Query (using DB2 CLI adapter) reports to Query/400 *QRYDFN are positive. Even with running to a client, the DB2 Web Query reports run faster, use less much less CPU, and require about the same amount of memory. This can primarily be attributed to the DB2 CLI adapter generating SQL and using the new SQL Query Engine (SQE).
- ▶ Comparisons of Web Query to SQL show that DB2 Web Query has longer response times, which again is expected when running to a client, and uses more resources. The CPU used by Web Query is two to four times the CPU used when running SQL, depending on whether the full open re-optimization was forced every time. Note that Web Query recycles the server jobs for every query.
- ▶ If you are using DB2 Web Query to modernize Query/400 queries (by creating new reports), there is little additional overhead.
- ▶ If you are creating new reports, always keep in mind that SQL is being generated. Therefore, an understanding of SQL performance basics is vital.



Frequently Asked Questions

In this chapter, we list frequently asked questions (FAQs) regarding DB2 Web Query. We have organized the questions into the following categories:

- ▶ “General announcement or product information questions” on page 342
- ▶ “Installation and setup questions” on page 344
- ▶ “Security-related questions” on page 345
- ▶ “Named-user question” on page 346
- ▶ “Metadata questions” on page 347
- ▶ “Query/400 migration questions” on page 350
- ▶ “Report development questions” on page 351
- ▶ “Excel spreadsheet integration questions” on page 356
- ▶ “Add-on product questions” on page 358
- ▶ “Save and restore questions” on page 360
- ▶ “Education questions” on page 361

15.1 General announcement or product information questions

In this section, we address general questions regarding the announcement and product information.

What was announced?

IBM is delivering a new Web-based query and report writing product that offers enhanced capabilities over the IBM Query for iSeries (also commonly known as Query/400) product (5722-QU1). The new product, IBM DB2 Web Query for System i (5733-QU2), provides several easy-to-use tools for building business intelligence solutions, including reporting and graphing assistant components, a WYSIWYG report “painter,” and the optional Developer Workbench for creating more customized reports.

You can extend DB2 Web Query into multidimensional analysis with the optional online analytical processing (OLAP) module or provide reports for users on the go with the Active Reports option. You can easily modernize existing Query for iSeries definitions through an import function and then enhance them with the Web-based tools.

What are some of the capabilities of this new product?

DB2 Web Query provides capabilities to query or build reports against data that is stored in DB2 for i5/OS databases through the latest browser-based user interface technologies. You can build new reports with ease through the Power Painter, Report Assistant, and Graph Assistant components. You can also simplify the management of reports by leveraging parameterized reporting. In addition, you can deliver data to users in many different formats, including spreadsheets, PDF, and HTML or through the Java-based thin client interface browser support.

DB2 Web Query offers features to import Query/400 definitions and enhance their look and functionality with Power Painter or Graph Assistant. You can add OLAP-like slicing and dicing to the reports or view reports in disconnected mode for the user who are on the go. In addition, you can interface to all data in i5/OS through either DB2 or Open Query File native adapters that automatically identify the files to be accessed and import the metadata into DB2 Web Query.

We are happy with our Query/400 reports. Is there any reason to move to this new product?

There are several reasons why you might want to consider using this new technology:

- ▶ Query/400 does not leverage the latest query optimization technology in DB2 for i5/OS, which is called SQL Query Engine (SQE)). DB2 Web Query offers the ability for the queries to use the query optimization features that have been added to DB2 for i5/OS over the last few years. DB2 Web Query leverages the improved performance, automation, and monitoring capabilities of SQE.
- ▶ The parameterized reporting feature of this new product can substantially reduce the number of reports that must be maintained individually, reducing the backlog of report writing requests.
- ▶ Today's requests for information demand that the data be delivered in many different formats and on a request, scheduled, or alert basis. Information must be provided via easy to interpret graphical representations, dashboards, or scorecards or dumped into spreadsheets. Query/400 is limited in its ability to meet these demands.

Why a Web-based query and report writing function? Is a Windows-based solution not more robust?

The product is based on the latest Web 2.0 and AJAX Web technologies. It allows you the functionality and usability that is usually associated with a Windows environment, but is now available to browser-based applications.

The zero footprint solution reduces complexity and costs. It also allows any enterprise browser to be leveraged as the user interface. The AJAX framework allows sophisticated report building capabilities including modeless and modal dialogs, menus, multiple coordinated windows, partial panel updates, and more.

A Web-based solution offers the following benefits:

- ▶ Lower installation costs
- ▶ Lower maintenance costs
- ▶ Easy, incremental upgrades to existing applications
- ▶ User administration savings
- ▶ Simpler and more controlled document backup and archiving
- ▶ Simpler and more controlled security and compliance administration
- ▶ Software updates controlled on the server making them available to all users concurrently

What additional features are available with DB2 Web Query?

The following additional capabilities are available that leverage the foundation set by DB2 Web Query:

▶ Active Reports

The Internet makes vital information easily accessible to large numbers of people. But what about mobile workers who are frequently disconnected from the Web, such as sales people who spend most of their time visiting clients or service professionals who travel from customer site to customer site?

With the DB2 Web Query Active Reports feature, users like these can take the power of business intelligence with them wherever they go. By combining data and interactive controls into a single, self-contained HTML file, Active Reports deliver analytic capabilities in a completely portable and disconnected environment, with absolutely no software required. Users can manipulate reports in various sort orders, filter data by different criteria, and chart information for visual impact—anytime, anywhere.

▶ OLAP module

Business analysts know that even a slight variation in the way they look at data often uncovers previously indiscernible trends and relationships and produce remarkable insights. If each new view requires a new report request, more time is spent building and processing reports than analyzing results and taking the appropriate action. The OLAP feature solves the data analysis issues that business analysts face by adding in-depth, built-in analytical processing to the product's robust and scalable enterprise reporting and information delivery capabilities.

Note: In order to leverage the OLAP functionality, metadata that defines data relationships is required. This metadata is created with Developer Workbench. Therefore, we recommend that either you have a license of Developer Workbench or another mechanism, such as an ISV-supplied metadata layer, provides the metadata.

▶ Developer Workbench

The DB2 Web Query Developer Workbench feature is an open and intuitive environment that allows for rapid development of more customized Web-based reports. It includes an

HTML layout painter for building compound reports or dashboards, combining multiple reports onto a single view.

The Developer Workbench's synonym editor is used for building metadata to hide complexities of the data from end users, such as decomposing date fields, applying pre- and post-calculation rules, or any of the hundreds of data manipulation functions. You can use the metadata interface to define relationships in the data, such as defining the hierarchy of dimensions that automates the drill-down function of the OLAP feature.

Is there any integration with the Rational tools with this product?

No, there is no IBM WebSphere® or Rational® integration.

Is there integration to portlets with Portal 6.0?

V1R1 of DB2 Web Query allows you to launch the signon window by clicking a URL from the portal, but it is not tightly integrated at this point. This is a known requirement.

Can a model 8nn system have the product QU2 installed?

Any system that has a supported release of V5R4 can have the product ordered and installed.

Do I need to download data to a PC server or install a Windows server to sit between the client and the System i server?

Many enterprise reporting tools on the market force you to move your data out of DB2, or require multiple servers to support various reporting functions. The IBM product is an i5/OS and thin client implementation, which simplifies the whole reporting infrastructure for System i customers. Software upgrades are simplified, reliability can be increased, network traffic is reduced, and security and auditing can be simplified.

15.2 Installation and setup questions

In this section, we answer questions regarding the installation and setup of DB2 Web Query.

I noticed that DB2 Web Query is using port 11331 for HTTP requests. What if I am already using this port for something else? Is there any way to change it?

DB2 Web Query demands the exclusive use of ports 11331 through 11339. It is possible to change the port. However if the reason you want to change the port is due to port number conflicts with another application, we strongly recommend that you change the other application to use a different port outside of that range. If that change is not possible, IBM can provide documentation on how to change the ports used by DB2 Web Query on a case-by-case basis. However, doing so can limit the ability of IBM to provide support for DB2 Web Query. If this is your only option, contact IBM support for more details.

When I access the DB2 Web Query web console, I only see a limited number of options. What happened to the other options?

If you sign into the DB2 Web Query web console (http://systemname:11331/webquery_html/wfconsole.htm) and only see limited configuration options, you are probably not signed in as the user profile QWEBQRYADM. Only this profile is able to see and use all of the web console configuration options

Is the server-based processing performed by the WebSphere Application Server or something that requires less overhead?

A lighter weight integrated application serving environment is used, one with more specific and limited functionality. The product does not require any WebSphere Application Server instances to be running.

How are the DB2 Web Query domains established?

Only an administrative class user can create new domains. Such a user must be a member of the group profile MRADMIN. Domain creation is a manual process that can only be performed by members of this group profile.

Can I configure DB2 Web Query to start automatically after an IPL?

Yes, you can. However it is important that the STRWEBQRY command always be run in a job with user profile QWEBQRYADM. By default, the system startup program QSTRUP runs with user profile QPGMR. To make sure that QSTRUP can start DB2 Web Query properly:

1. Enter the following command and add user QPGMR with object authority *USE:
`EDTOBJAUT OBJ(QWEBQRYADM) OBJTYPE(*USRPRF)`
2. In the CL program QSTRUP, add the following line to start DB2 Web Query:

Attention: TCP must be active before you enter the STRWEBQRY command.

```
SBMJOB CMD(QWEBQRY76/STRWEBQRY) USER(QWEBQRYADM)
```

15.3 Security-related questions

In this section, we address questions regarding security.

Given that this is browser-based access, how is security established for users?

First, users must log into DB2 Web Query using their i5/OS user profiles and passwords. If that user has been registered as a named user, access is allowed to the DB2 Web Query launch page.

From the launch page, a user is allowed to work with reports within a DB2 Web Query domain. Each domain is associated with two group profiles: one for report developers and one for report users. User profiles are then added to the appropriate group profile. If a user profile is a member of the “report developer” group profile associated with the domain, that user can create, edit, and run reports within that domain. Similarly, user profiles that are members of the “report user” group profile can (only) run all reports within the associated domain. The developer class named users are members of a group user profile name prefixed by MD, and user class named users are members of a group user profile name that is prefixed by MU. For example, if your domain is named SALESREP, the developer group profile is named MDSALESREP, and the user level group profile is MUSALESREP.

In addition, when the report is executing, normal i5/OS object-level security is enforced. Therefore, if the user who running the query does not have the appropriate authority to the underlying database object, the query execution fails accordingly.

Can my DB2 Web Query reports run with adopted authority?

Yes. If you need the ability to run with adopted authority, you must register an existing program (one with adopted authority) as a stored procedure. You then create a synonym against that stored procedure. In order for DB2 Web Query to receive data back and create a report against that data, the program or stored procedure (or one in the subsequent call stack) must return at least one result set.

My user profiles are already members of a group profile. Since this user profile parameter (GRPPRF) is already specified, how can domain authority be established?

Use the Supplemental Group (SUPGRPPRF) parameter of the CRTUSRPRF/CHGUSRPRF command. This parameter accepts multiple values and can also be used to establish the association of a user with a domain.

Can a more granular level of security be established, for example, to limit a report for a specific user?

From the perspective of which reports are displayed to the user on the DB2 Web Query launch page, the answer is “No”. Domain level is the lowest level of security and determines which reports are displayed to a particular user. However, object-level security that is implemented in the System i5 environment is honored. This means that users might see a report. However, if they are not authorized to the underlying database objects used by the report, the report execution will fail.

Is sign-on to the product available via LDAP or Kerberos?

No, there is currently no support for single sign-on environments such as Lightweight Directory Access Protocol (LDAP) and Kerberos.

Is auditing available for running a DB2 Web Query report?

DB2 Web Query does not have an auditing tool, but DB2 for i5/OS and i5/OS have tools such as the use of stored procedures for which specific logging can be done, the database open exit program, the SQL call-level interface (CLI) exit program, and object journaling.

15.4 Named-user question

In this section, we provide answers to questions regarding named users.

How are named users managed, established, or removed?

The DB2 Web Query product provides programs that are interfaces to the licensing APIs to both add and remove a user. Thus the named users are tracked in the licensing information:

- To view the named users, enter the following command:

```
WRKLICINF PRDID(5733QU2)
```

The select Option 8 for feature 5050.

- To add a named user, enter the following command:

```
CALL QWEBQRY76/WQADDLIC 'USRPRFNAME'
```

- To remove a named user, enter the following command:

```
CALL QWEBQRY76/WQRLSLIC 'USRPRFNAME'
```

Note: The value of the user profile parameter of both the WQADDLIC and WQRLSLIC programs must be enclosed in single quotation marks and must be in uppercase.

15.5 Metadata questions

Here we address questions regarding metadata.

If there is a known join relationship that is not defined to the database using DB2 for i5/OS referential integrity features, is there a way to define such a relationship in the metadata?

Yes, you can do this by creating synonyms from the Developer Workbench toolset. Refer to 12.1.4, “Joining tables” on page 275, which provides an example of how to define this relationship.

Alternatively, to define the relationship, you can create SQL VIEWS, add the referential integrity definitions at the database level, or manually add the Join specifications for the reporting.

Is there a way to automate the generation or refreshing of the metadata, for example by using an API?

No, there is neither metadata automation nor an API to build or rebuild the metadata. It must be done manually. Therefore, if you already have metadata against a table in place, and that table has been altered (for example, new columns have been added), you must manually refresh the table's metadata before the new columns can be used in your reports.

How do I manage my synonyms?

Synonyms are stored in the integrated file system (Q\IBM\UserData\webquery\ibi\apps\baseapp directory) and can be managed from one of the following interfaces

- ▶ Windows Explorer

You must first set up a mapped network drive. From this interface, you can only delete existing synonyms.

- ▶ iSeries Navigator

From this interface, you can only delete existing synonyms.

- ▶ WRKLNK command

From this interface, you can only delete existing synonyms.

► Developer Workbench

From the Developer Workbench client, you can do much more than simply deleting the synonyms. By right-clicking the selected synonym, you see the list of available options as shown in Figure 15-1.

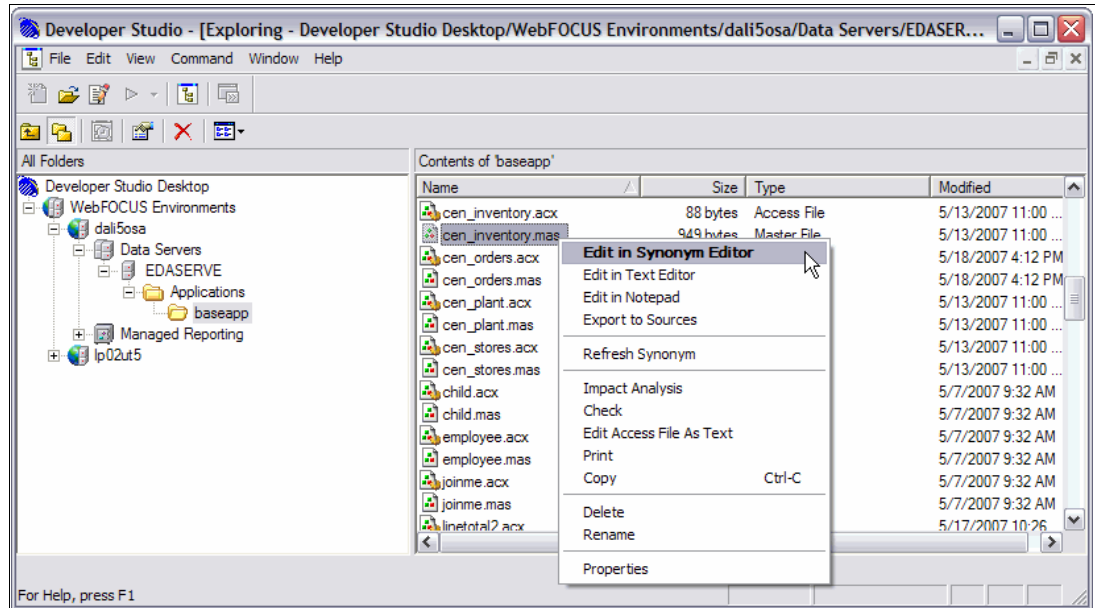


Figure 15-1 Manage synonyms from Developer Workbench

The following options are some of the more useful options that are available through this interface:

– Synonym Editor

The Synonym Editor provides exclusive features such as:

- The ability to decompose date fields.
- An interface for defining filters, virtual fields, and business views.
- The ability to define appropriate dimensions for the OLAP module.

– Refresh Synonyms

Refreshing a synonym enables you to update field information while preserving the original synonym title, description, usage, virtual field, and database administrator (DBA) information. The action also synchronizes the Master File with the table on which the synonym is based.

– Domain-level report movement

Use the drag features to move your domains from one system to another.

Within my reports, how can I simulate a library list-like behavior when accessing the base tables and views?

During the report development and testing phases, you might prefer to create your reports against tables or views in a test library, verify that the reports are correct, and then at some point, change the report to point to data in a different (perhaps production) library. Typically, this is implemented easily from many System i interfaces by using and manipulating your library list. However, by default, your synonym is based on a table or view that is selected from a specific library; therefore, the library list behavior does not occur. Creating separate

versions of the report for each data library is one approach to consider, but is likely not desirable or practical if you have many reports to create and maintain.

To obtain the library list behavior and avoid creating a separate report for each data library, you must select the *One part name* synonym setting as shown in Figure 15-2.

Create Synonym for DB2 cli (*LOCAL) Step 2 of 2

☐ Cardinality ☐ With foreign keys ☐ Dynamic columns ☒ One-part name

Select application: baseapp Prefix: century_ Suffix:

☐ Customize data type mappings

☐ Overwrite existing synonyms Create synonym

	Default synonym name	Library/Schema	Table name	
☐	ALERTS	CENTURY	ALERTS	TAB
☐	CURRRATE	CENTURY	CURRRATE	TAB
☐	HR	CENTURY	HR	TAB
☐	INVENTORY	CENTURY	INVENTORY	TAB
☒	ORDERS	CENTURY	ORDERS	TAB
☐	PLANT	CENTURY	PLANT	TAB
☐	STORES	CENTURY	STORES	TAB

Figure 15-2 Specifying one-part name during synonym creation

Specifying one part name for the synonym prevents DB2 Web Query from hard coding the library name in the meta data. When the report (based on a one part name synonym) is run, the requesting user’s library list is searched for the table or view names on which the synonym is based. This means that you must either log in as a different user profile in order to have the report access data in a different library, or use an easier technique to change the library list of the job description that is defined to the user profile.

As long as the formats of the files in the different libraries are the same, this method is possible to carry out.

Note: From a DB2 Web Query perspective, the user’s library list is determined and controlled by the system values QSYSLIBL and QUSRLIBL as well as the job description that is defined for the user profile. Any other method of setting the library list is not recognized by DB2 Web Query. For example, if you are using the initial program (specified in the user profile) to manipulate the library list when the user signs on, this method will fail because the initial program is not executed during the DB2 Web Query session.

Can I query DB2 for i5/OS databases, either on another partition or in another server, without requiring multiple licenses of DB2 Web Query on each of these partitions or servers?

With additional adapters as add-ons to DB2 Web Query, you can build a “federated” query that combines data from many different back-end databases into a single DB2 Web Query report. Over two hundred adapters are available to connect to other database or ERP systems.

You can also combine data from multiple DB2 for i5/OS databases running on different servers or logical partitions (LPARs), without requiring any additional adapters since the base product ships with the DB2 for i5/OS adapter. For more information about setting up a cross-system join, see 3.4.2, “Setting up a cross-system join” on page 45.

15.6 Query/400 migration questions

In this section, we address questions regarding Query/500 migration.

What happens to my existing queries when I upgrade to 5733-QU2?

Nothing happens to your existing queries when you upgrade to 5733-QU2. They still run and the Query/400 product is still supported.

Attention: Do not remove the Query/400 licensed program product (LPP) 5722-QU1. This LPP is still required, even after you install 5733-QU2.

How do I migrate Query/400 reports into DB2 Web Query?

Migration of queries is accomplished via an automated facility that allows you to select the query or queries that you want to translate and produces the new Web Query report ready to run or to be brought into one of the new user query components for customization. For more information about migrating Query/400 reports, see 13.1, "Query/400: A reliable reporting tool" on page 302.

Is there an automated way to import Query/400 objects?

No. The process to import queries is a manual one.

Query/400 supported exception joins. How do I specify that type of join with DB2 Web Query?

The DB2 Web Query development tools available through the Web browser interface (Report Assistant, Graph Assistant, and Power Painter) do not support the specification of left exception, right exception, and right outer join types. If you need this functionality in your report, you must take the following steps:

1. Create an SQL view with the appropriate tables and join syntax. Here is an example of creating a view with an exception join:

```
CREATE VIEW century/examplejoin AS (  
  SELECT a."StoreName", b."LineTotal"  
  from century/stores a LEFT EXCEPTION JOIN century/orders b  
  ON a."StoreCode" = b."StoreCode")
```

Here is an example of creating a view with a right outer join:

```
CREATE VIEW century/examplejoin AS (  
  SELECT a."StoreName", b."LineTotal"  
  from century/stores a RIGHT OUTER JOIN century/orders b  
  ON a."StoreCode" = b."StoreCode")
```

2. Create a synonym against that view.
3. Base your report on the synonym of the view.

Will the imported Query/400 definition start using SQL, so that the SQL Query Engine versus the Classic Query Engine will be used?

No. The reports based on Query Definition objects are generated from the metadata that is available from the query object itself and will continue to be run using the RUNQRY which uses the Classic Query Engine (CQE). A new Web Query should be created to replace an existing query, to enable an SQL interface to the data. This is the only way to have the query processed by the SQE.

Will a Query/400 object that was previously imported still function as an imported report, even after the original query definition has been changed?

Yes, the object will still function as an imported report as long as the definition of the original Query/400 object is not changed to remove fields or existing fields are changed to an incompatible type with those in the existing metadata.

Can an imported Query/400 query be saved with removal of reference to the existing query; effectively creating an actual Web Query report from the imported query?

No. To fully detach the reference to the original query definition, a new DB2 Web Query report must be created without reference to metadata from a query definition object.

Some Query/400 definitions are scripted in a CL program, so that one query produces output that will be used by the next query for input. Will that be possible with DB2 Web Query?

You can accomplish the same thing, but use of the QTEMP library might be an inhibitor. The browser initiated work runs in the QSQSRVR jobs, and there is no assurance that the QTEMP for one query will be the same as for the next query. A permanent library must be specified.

15.7 Report development questions

Here, we answer questions regarding report development.

I have my own style sheets that I want to use for my reports. How do I make them available from Report Assistant?

In Report Assistant, on the Reporting options tab, you can specify a style sheet under the Apply an existing WebFocus Stylesheet field. To add your own style sheets to this list:

1. Copy your stylesheet files to the integrated file system directory
\\QIBM\\UserData\\webquery\\ibi\\webfocus76\\basedir\\import.

After they are in that directory, they are available to every domain for importing.

2. To import, open the desired domain, right-click the **Other Files** folder, and select **Import**.
3. In the Select Files to Add dialog window that opens, from the list of style sheets, select the stylesheet files and click **Save**.

The style sheets are now available from Report Assistant.

Tip: Many sample style sheets are provided as part of the base product and can be used in your reports. They have the file extension .STY and can be found in the integrated file system directory \\QIBM\\UserData\\webquery\\ibi\\webfocus76\\template.

To make these style sheets available in the Report Assistant interface, copy the .STY files to the directory specified in step 1 above and follow the remaining steps.

Attention: Although custom style sheets should work in DB2 Web Query, IBM Support does not provide assistance if you should experience problems.

How can I limit the number of rows that are retrieved during the report development phase?

If you are developing reports that are retrieving large numbers of rows from the database, you might find it useful to restrict the number of rows that are returned and avoid waiting for all this information to be returned to the client.

To obtain this behavior:

1. Open the report.
2. Select **Report options** tab.
3. As shown in Figure 15-3, in the Content and generation pane, specify the number of records that you want retrieved.
4. Click **Save**.

Important: Do not forget to remove this setting prior to moving your report into production. Failure to do so results in a report that does not retrieve all of the rows that satisfy the selection criteria of the report.

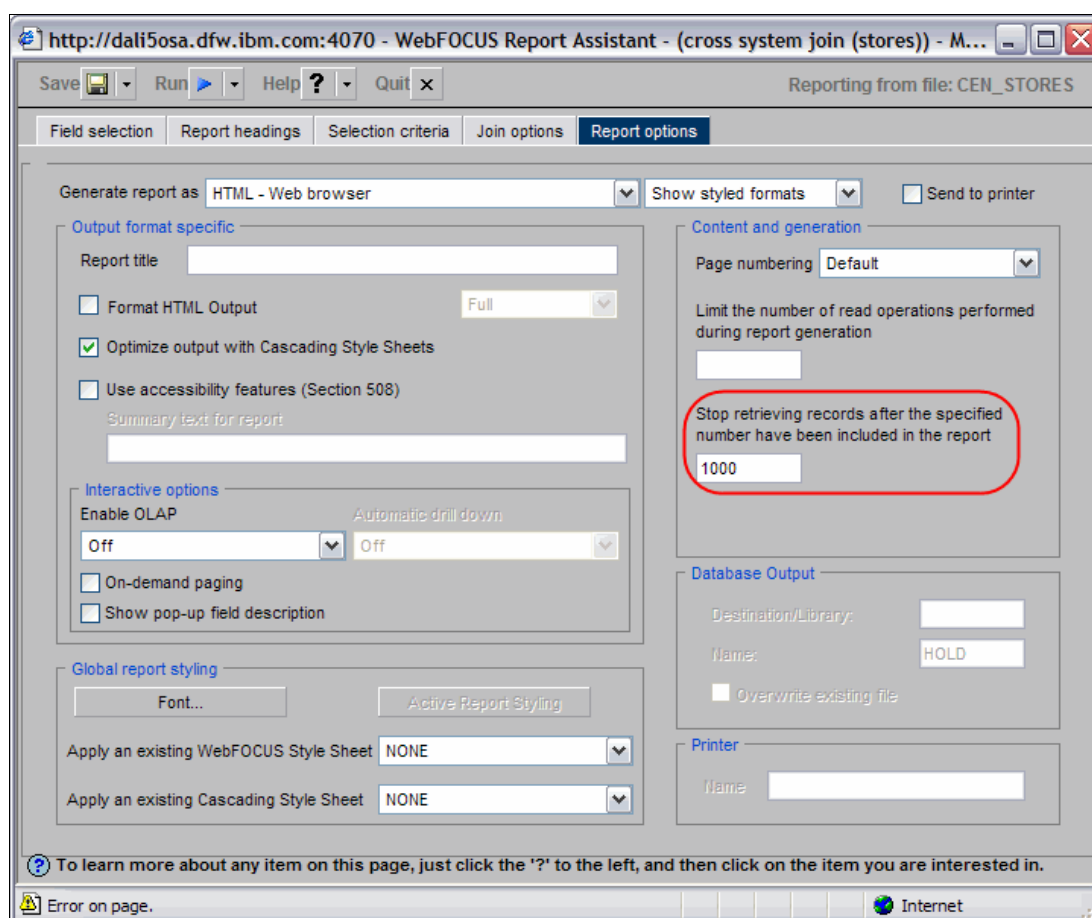


Figure 15-3 Limiting the number of rows retrieved

Can DB2 Web Query handle stored procedures with input parameters?

Yes, DB2 allows your stored procedures to have input parameters. However, you must specify a value for the input parameter during the synonym creation process. Although it must be specified here, this value is not stored with the synonym definition and is not passed to the procedure during execution.

When creating the report against the stored procedure, you see the input parameter listed in the list of available fields. From the Selection criteria tab, select this field as shown in Figure 15-4.

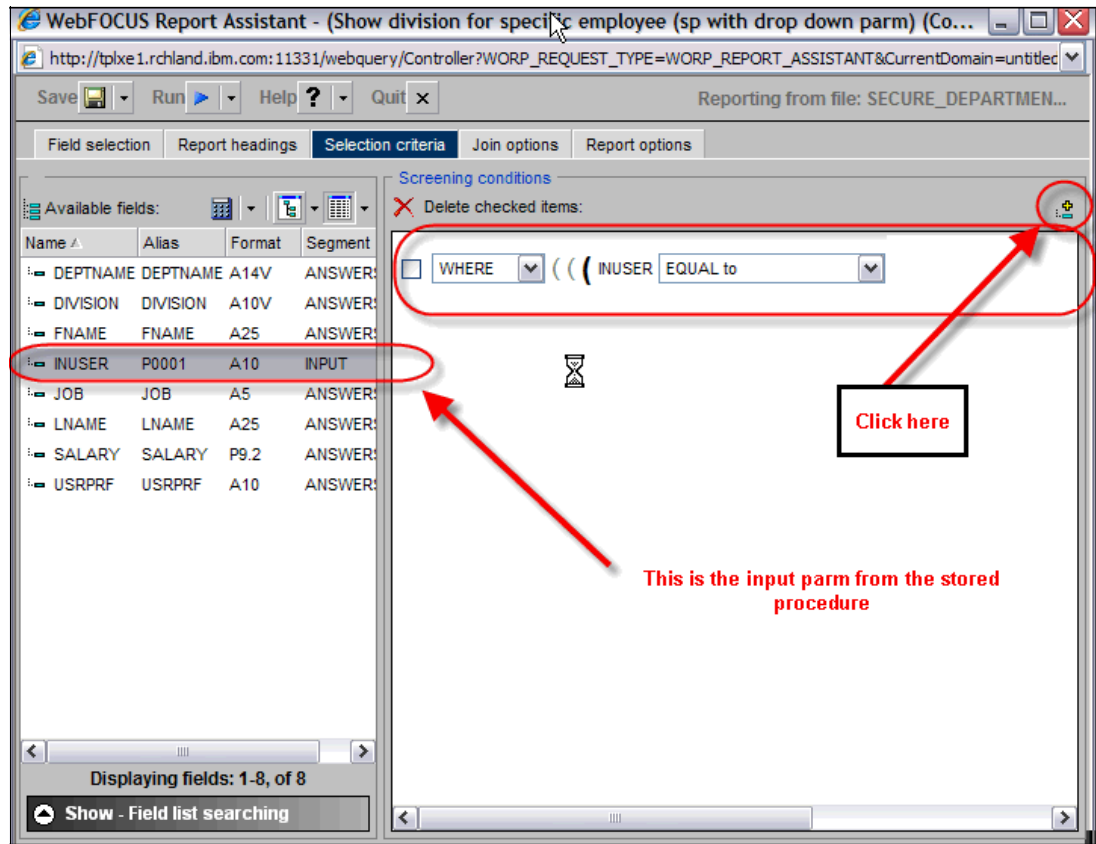


Figure 15-4 Using a stored procedure with an input parameter

If you want the input parameter that is passed from the report to the stored procedure to be selected from a list, you must specify a different source synonym than the stored procedure.

How can the SQL that is produced for the report be viewed?

You can view the SQL by using the *Run w/SQL* option. This option sends the SQL statement generated by DB2 Web Query to the browser. In addition, the i5/OS database features in iSeries Navigator, such as the SQE Plan cache and Database Monitor, can be used to determine which SQL statement was produced and submitted to the database engine.

In parameterized reports, can more than one item be selected from the list?

Yes, more than one item can be selected as long as the parameter was defined to allow multiple selections. By using the standard Windows selection method, press Ctrl and click to add or remove a selected entry.

Is there a way to specify double-spaced printing?

Yes, double-spaced printing is controlled by custom style sheets.

Can a drill-down report have further drill-downs defined?

The active link for the drill-down is just another report. As many levels exist for which a report can represent another drill-down. The only limit is the existence of the report to process the link.

However if the parent report is in the output format of an Excel spreadsheet, drill down only works if the report is displayed via Excel embedded in the browser window. If you have configured your browser to launch the Excel application automatically when opening the spreadsheet report, the drill-down capability is disabled.

Can I create dashboards by using the Power Painter tool?

Power Painter enables you to specify multiple reports to appear on a single page. However Developer Workbench provides an HTML Layout Painter that is typically used to design a dashboard for an HTML interface such as a Web browser.

When building a dashboard, can graphics be included, for example a company logo?

Yes, you can include graphics when building a dashboard by using the HTML Layout Painter tool (part of Developer Workbench) or Power Painter. The dashboard is a Web page, which can have a large variety of elements placed for viewing.

Will the dashboard allow a periodic report, an effective automatic update?

No. The report must be obtained or refreshed manually.

Can I run a DB2 Web Query report from within a CL program?

Yes, the RUNWEBQRY command is delivered with the product and provides support for spooled file and database output options. Figure 15-5 shows the format of the RUNWEBQRY command.

RUN JAVA BATCH IN WEBQUERY (RUNWEBQRY)		
Type choices, press Enter.		
DOMAIN NAME, OR HREF	UNTITLED	Character value
FOLDER NAME, OR HREF		Character value
REPORT (FEX) NAME		
OUTPUT FORMAT	WP	DOC, WP
OUTPUT QUEUE	QPRINT	Character value
RUN WITH USER ID	*CURRENT	Character value

Figure 15-5 RUNWEBQRY command

In order to run a report using this command, you must specify the appropriate domain and folder HREFs of the desired report. To obtain this information:

1. Open a DB2 Web Query session using your Web browser.
2. Expand the tree until you see the report. Right-click the domain that contains the report and select **Properties**.
3. In the Properties window that opens, you see the Href value, which you specify in the DOMAIN parameter of the RUNWEBQRY command. For example, let us assume that you see the following value:

db2wbqry/db2wbqry.htm

In this case, you specify the value db2wbqry in the DOMAIN parameter.

4. Right-click the folder that contains the report and select **Properties**.
5. In the Properties window, you see the Href value. This is the value that you specify in the FOLDER parameter of the RUNWEBQRY command. If the Href has a leading number (#) character, do *not* include this in the parameter value. For example, let us assume that you see the following value:

#visiblea56gb

In this case, you specify the value visiblea56gb in the FOLDER parameter.

6. Right-click the report and select **Properties**.

You see the Filename value in the displayed Properties window, for example:

app/newReport.fex

Excluding the leading “app/” and trailing “.fex” characters, this is the value that you specify in the REPORT (FEX) NAME parameter of the RUNWEBQRY command. Therefore, in the case of the above example, the value specified is newReport.

Putting it all together and using the example values that we provided, your command looks like the example shown in Figure 15-6.

```

                                RUN JAVA BATCH IN WEBQUERY (RUNWEBQRY)

Type choices, press Enter.

DOMAIN NAME, OR HREF . . . . . > db2wbqry      Character value
FOLDER NAME, OR HREF . . . . . > visiblea56gb   Character value
REPORT (FEX) NAME . . . . . > newReport

OUTPUT FORMAT . . . . . WP                      DOC, WP
OUTPUT QUEUE . . . . . QPRINT                   Character value
RUN WITH USER ID . . . . . *CURRENT             Character value
```

Figure 15-6 RUNWEBQRY command with values specified

When I prompt (F4) the RUNWEBQRY command, I do not see “Database file” as an option for output format. I thought this was one of the supported output formats.

The RUNWEBQRY CL command does support output to a database file. This output option must be defined in the report itself. If you have “Database file” specified in the DB2 Web Query report definition, this setting is always honored in the RUNWEBQRY command, even if you specify a target output queue in the command’s OUTQ parameter. Any other output

format specified in the report definition results in the report output being sent to a spooled file when the CL command is invoked.

Do I have to be registered as a named user before I can use the RUNWEBQRY command?

Yes, use of the RUNWEBQRY command requires the submitting user to be registered as a DB2 Web Query named user. However, if you are not a named user and you have sufficient authority to submit a job on behalf of another user profile, you can specify a valid named user in the USER parameter of the RUNWEBQRY command. When you do this, the batch jobs submitted for this request run under the user profile that you specified. If this user profile is a valid DB2 Web Query named user, the request is allowed to continue.

Can report selection be defined on an expression?

In many cases, report selection can be defined on an expression by using defined fields. There are limitations, such as expressions that use aggregate functions, for example, percentage of total. In these cases, you can use an SQL view or the SQL wizard to create a report with these requirements.

How is a parameterized report designed, specifically to establish the inputs?

The parameterized query defines the inputs using the HTML Layout Painter in the Developer Workbench.

15.8 Excel spreadsheet integration questions

For the following questions, we provide answers regarding Excel spreadsheet integration.

How do I open my Excel spreadsheet reports in Microsoft Excel instead of the Web browser?

By default, whenever you open a report whose output format is defined as an Excel spreadsheet, your browser attempts to open the XLS file in the same browser session (using an XLS plug-in). If you prefer to have the spreadsheet opened in Excel:

1. Open Windows Explorer.
2. Select **Tools** → **Folder Options**.
3. Select the **File Types** tab (Figure 15-7 on page 357). On this tab, complete these steps:
 - a. In the Registered file types list, find and select **XLS** (Microsoft Excel Worksheet).
 - b. Click the **Advanced** button.

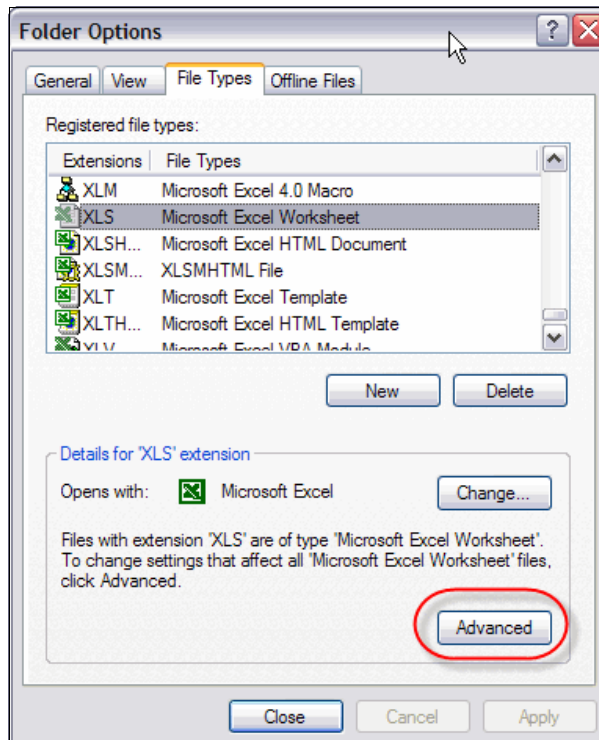


Figure 15-7 Windows folder options - Advanced

- c. In the Edit File Type window (Figure 15-8), make sure the “Confirm open after download” and “Browse in same window” options are *not* selected. Click **OK**.

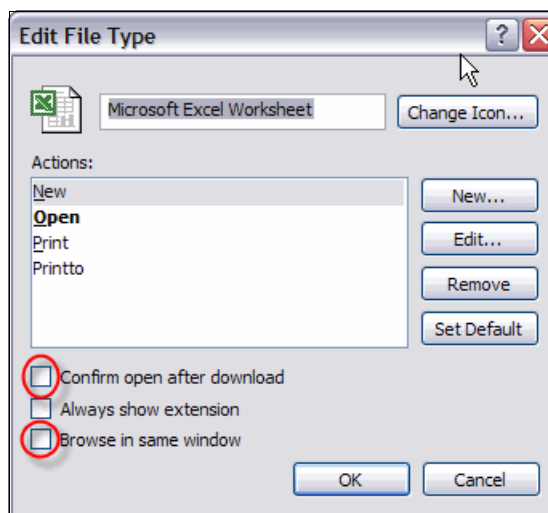


Figure 15-8 Edit File Type

- d. In the Folder Options window, click **Close**.
4. Close the Windows Explorer window and try again.

Attention: Be aware that when you configure your browser to behave this way when opening an Excel spreadsheet, you reports lose the ability to drill down to other reports.

I have created a report with output to an Excel spreadsheet. Does drill down (clicking an active link) access another spreadsheet?

Yes, drill down accesses another spreadsheet, but only if the spreadsheet is opened in the browser window. The drill down is active while in a browser, using an Excel plug-in to place the data from the referenced report in the browser window.

One of the report output options is “Excel with formulas.” What does that really mean?

Calculations and summaries are output to the Excel spreadsheet as formulas instead of literal values. This way the recipient of that spreadsheet can do row or column manipulations, for example delete a row, and the results of cells with formulas are automatically recalculated and updated.

Does DB2 Web Query support integration with Microsoft Excel 2007?

Yes, Excel 2007 is supported.

15.9 Add-on product questions

In this section, we answer questions regarding add-on products.

The OLAP feature is enabled by defining dimensions for the database. How does DB2 Web Query get those dimensions?

The dimensions must be defined in advance using the Developer Workbench.

When cascading through data in an OLAP report, is the data limited based on prior selections?

Yes. For example, specifying country as United States limits the States/Provinces list to only the states in the United States, so that no data associated with Canada is available to cascade in the list.

In an OLAP report, can the columns be sorted on demand (as they can be in Active Reports?)

Yes, you can re-sort the report, based on any of the displayed columns.

Does a copy of the Developer Workbench come with the base product?

No. The Developer Workbench is a feature of the IBM product, but is not part of the base product.

How large is an Active Report file?

The size of an Active Report file depends almost entirely on the amount of data from a file. The definitional attributes to allow actively working with the data in the browser will be relatively fixed in size, by comparison to the amount of actual data included in the report.

Can an Active Report be saved with the active visualizations and any other changes made by the user?

Yes. The Active Report file can be saved with the current formatting and visualizations specified by the user.

I understand that Active Reports can be sent to others via e-mail. In order to run the report, do the e-mail recipients have to be named users as well?

No. Only the user that originally ran the report and generated the Active HTML output needs to be a named user. After the Active HTML output file is created, it can be distributed to an unlimited number of recipients, none of which have to be named users (or even connected to the System i environment). The recipients open the HTML file using a browser and can work with the Active Report in the same way as the original (named) user that created the HTML output can.

How can I execute reports in batch and distribute automatically in PDF form via an e-mail distribution list?

In V1R1 of DB2 Web Query, this takes some effort because PDF format is an output option only by running the report in non-batch mode, and there is no automatic report distribution. This is a known requirement for the future.

Can a report developer create reports based on manually entered or imported SQL statements?

Yes, the SQL wizard feature of Developer Workbench provides an interface from which report developers can manually enter or copy and paste SQL statements, as well as point to SQL script files that exist on the integrated file system. As long as these SQL statements contain valid SELECT statements, they can be used as the basis for DB2 Web Query reports.

Why am I having trouble saving my Active Reports so that I can redistribute them?

First of all, you must be using a browser that supports ActiveX technology. This means that only Internet Explorer can be used if you want to save your Active Reports to your hard drive. Firefox has no native ActiveX capability, and consequently, cannot save the HTML file.

Second, you must make sure that your Internet Explorer browser settings are configured to enable ActiveX for Internet Explorer (version 6.x and 7.0). To set this up correctly:

1. In the browser, from the menu bar, select **Tools** → **Internet Options**.
2. In the Internet Options window, click the **Security** tab.
3. On the Security tab, click the **Custom Level** button.
4. In the Security Settings window, scroll through the menu and Select the **Enable** radio button for the following menu choices:
 - Run ActiveX Controls and Plugins
 - Script ActiveX Controls Marked Safe for Scripting
 - Initialize and Script ActiveX Controls not Marked as Safe for Scripting
5. Click **OK** to exit and save.
6. Click **OK** again to exit the Internet Options window.
7. Close and re-open the browser.

Active X is now enabled and you can save your Active Reports.

15.10 Save and restore questions

In the following questions and answers, we provide guidance about save and restore.

What is the recommended save and restore process for DB2 Web Query?

From a save and restore perspective, you must be concerned about the following components of DB2 Web Query:

- Synonyms (also referred to as metadata)

All synonyms are stored in the integrated file system directory QIBM\UserData\webquery\ibi\apps\baseapp. If you find that you are regularly creating or refreshing metadata information, we recommend that you save this directory frequently.

For example, to save all metadata into a savefile named SYNONYMS created in the QGPL library, you enter the following command:

```
SAV DEV('/QSYS.LIB/QGPL.LIB/SYNONYMS.FILE')
OBJ(('QIBM\UserData\webquery\ibi\apps\baseapp' *INCLUDE))
```

To restore this metadata on the target system, you specify the following command:

```
RST DEV('/QSYS.LIB/QGPL.LIB/SYNONYMS.FILE')
OBJ(('QIBM\UserData\webquery\ibi\apps\baseapp' *INCLUDE
'QIBM\UserData\webquery\ibi\apps\baseapp')) ALW0BJDIF(*ALL)
```

- Reports (also referred to as Focus Execs, FOCEXECs, and FEXs)

All reports are stored in the integrated file system directory QIBM\UserData\webquery\ibi\webfocus76\basedir\domain_name (where *domain_name* equals the name of the domain in which the report was created). To save all reports (distributed throughout the various domains), save the directory QIBM\UserData\webquery\ibi\webfocus76\basedir.

For example to save all reports in all domains into a save file named ALLREPORTS created in the QGPL library, you enter the following command:

```
SAV DEV('/QSYS.LIB/QGPL.LIB/ALLREPORTS.FILE')
OBJ(('QIBM\UserData\webquery\ibi\webfocus76\basedir' *INCLUDE))
```

Similar to the synonyms, we recommend that you save this directory periodically or more frequently if reports are created and updated on a regular basis.

To restore all the domains, folders, and reports to the target system, you specify the following command:

```
RST DEV('/QSYS.LIB/QGPL.LIB/ALLREPORTS.FILE')
OBJ(('QIBM\userdata\webquery\ibi\webfocus76\basedir' *INCLUDE
'QIBM\userdata\webquery\ibi\webfocus76\basedir')) ALW0BJDIF(*ALL)
```

Note: If you are looking for the report files in the Common (untitled) Domain, you can find them in the QIBM\UserData\webquery\ibi\webfocus76\basedir\untitled\app directory.

If you want to save both synonyms and reports in one step, simply save the directory QIBM\UserData\webquery\ibi as shown in the following example:

```
SAV DEV('/QSYS.LIB/QGPL.LIB/ALLSTUFF.FILE') OBJ(('QIBM\UserData\webquery\ibi'
*INCLUDE))
```

Can I move or deploy synonyms and reports individually?

Synonyms can be copied or moved individually using either Windows Explorer (and mapped network drives) or Developer Workbench (the recommended tool for this type of activity). Both the master file (.mas extension) and the access file (.acx extension) must be moved as a pair. Keep in mind that the tables, views, and stored procedures, for example, on which the synonyms are based must exist on the target system. Otherwise any reports that references the synonyms will fail to execute.

Individual reports cannot be copied or moved to another system. However, domain-level movement is supported. This means that all reports within a particular domain are moved or deployed together. Use the copy and paste or dragging capabilities in Developer Workbench to perform these tasks.

Note: When moving or copying domains to another system, problems might occur if your reports reference (via drill downs, hyperlinks, and so on) other reports or procedures that reside in another domain.

In addition, you can use Windows Explorer with mapped network drives to move domains:

1. Manually create the domain on the target system (from DB2 Web Query home page or Developer Workbench). The name of the target domain must match the name of the source domain.

Attention: Domain movement does not work from Windows Explorer if you skip step 1.

2. From Windows Explorer, map a network drive for both the source and target systems.
3. From Windows Explorer, drag the source domain directory to the target directory T:\QIBM\userdata\webquery\ibi\webfocus76\basedir (where T is the target mapped network drive letter).
4. In the Confirm Folder Replace window, you see the message "This folder already contains a folder named..." Click the **Yes to All** button.

15.11 Education questions

In this section, we refer you to resources for education regarding DB2 Web Query on the System i platform.

Where can I find more information about the product?

You can find more information about the DB2 Web Query for System i home page at:

<http://www.ibm.com/systems/i/software/db2/webquery/>

This Web site is updated with the latest information about the product, including recent APAR information, new announcements, and downloadable demonstrations. It also includes links to find more information about add-on components available from Information Builders. In addition, if you are interested in business intelligence solutions on the System i platform, you can find links that provide more information about this subject, including success stores of customers who have implemented business intelligence solutions on the System i platform.

Where can I find more training for DB2 Web Query?

Refer to the DB2 Web Query Web site at the following address and click the **Getting Started** tab:

<http://www-03.ibm.com/systems/i/software/db2/webquery/index.html>

In addition, System i eLearning training is available on the Information Builder Web site at:

<http://education.informationbuilders.com/edu/Systemi.html>

Registrants to the training have access to the site for 30 days. Following the 30-day trial, the service is available on an annual subscription basis for a nominal fee.

Appendixes

In this part, we explain more specific topics on DB2 Web Query. This part includes the following appendixes:

- ▶ Appendix A, “Metadata in the integrated file system” on page 365
- ▶ Appendix B, “Date and time functionality” on page 367
- ▶ Appendix C, “Processing differences between the define and compute fields” on page 375
- ▶ Appendix D, “Setting up DB2 Web Query to run in an IASP environment” on page 377
- ▶ Appendix E, “Additional material” on page 385



A

Metadata in the integrated file system

In this appendix, we provide a more detailed explanation of what is created on the System i5 environment when DB2 Web Query metadata is created.

Every time a synonym is created in DB2 Web Query, two files are automatically created in the integrated file system that allow DB2 Web Query to read from that object. These files are located in the directory '/qibm/userdata/ibi/apps/baseapp'. The two files that are created are called a master file and an access file:

- *Master file*: This file identifies the fields of the table or, if the object is a QRYDFN, the fields of that query's result set. The master file has an extension of .mas. The fields are described in terms of length and data type. This file is similar to the information in a DSPFFD file, but the information is described in a way in which DB2 Web Query understands. Figure A-1 shows a master file for table QWQCEN/ORDERS.

```

Browse : /qibm/userdata/ibi/apps/baseapp/cen_orders.mas
Record : _____1_____ of _____26 by _____14_____ Column : _____1_____ 72 by _____79_____
Control : _____

.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....
*****Beginning of data*****
FILENAME=CEN_ORDERS, SUFFIX=DB2 , $
SEGMENT=CEN_ORDERS, SEGTYPE=S0, $
  FIELDNAME=ORDER_NUM, ALIAS=ORDER_NUM, USAGE=A5, ACTUAL=A5,
  TITLE='Order,Nbr', $
  FIELDNAME=INV_CODE, ALIAS=INV_CODE, USAGE=A4, ACTUAL=A4,
  TITLE='Inv,Code', $
  FIELDNAME=ORDER_DATE, ALIAS=ORDER_DATE, USAGE=HYMDm, ACTUAL=HYMDm,
  TITLE='Order,Date', $
  FIELDNAME=LOC_CODE, ALIAS=LOC_CODE, USAGE=A6, ACTUAL=A6,
  TITLE='Location,Code', $
  FIELDNAME=PLANT_CODE, ALIAS=PLANT_CODE, USAGE=A3, ACTUAL=A3,
  TITLE='Plant,Code', $
  FIELDNAME=SALES_REP, ALIAS=SALES_REP, USAGE=A25, ACTUAL=A25,
  TITLE='Sales,Rep', $

```

Figure A-1 Master file for the QWQCEN/ORDERS table

- *Access file:* This file identifies the object name and type on which the metadata is created. The access file has an extension of .acx. If the object is a QRYDFN, it says QIQRV. If the object type is a table, it says TABLENAME. Figure A-2 shows an access file for table QWQCEN/ORDERS.

```

Browse : /qibm/userdata/ibi/apps/baseapp/cen_orders.acx
Record : 1 of 1 by 14 Column : 1 75 by 79
Control :

.....1.....2.....3.....4.....5.....6.....7.....
*****Beginning of data*****
SEGNAME=CEN_ORDERS, TABLENAME=CENTURY/ORDERS, CONNECTION=*LOCAL, KEYS=0, $
*****End of Data*****

```

Figure A-2 Access file for the QWQCEN/ORDERS table

For QRYDFN objects, there are three additional metadata files: .fex, .txt and .inf.

Deleting metadata

There is no interface in DB2 Web Query to delete metadata. If you know that there are no more reports based on this object, nor will anymore be written, you can manually delete the metadata.

1. From the System i command line, enter the following command:
WRKLNK '/QIBM/USERDATA/IBI/APPS/BASEAPP'
2. Select option 5 by BASEAPP.
3. Browse the BASEAPP directory for the metadata that you want to delete. When you find it, select option 4 and press Enter to delete.

If you change the underlying table or QRYDFN, and if you recreate the metadata with the “overwrite” option, the master file is correctly updated. A new master file is not created. Therefore, you do not need to manually delete any metadata if you only used the “overwrite” option.



B

Date and time functionality

Date support is strong in DB2 Web Query. You can find most of the details in this appendix in the help text that is provided with Developer Workbench. Because the information is not in the help text for the base product, we have reproduced it here.

Date and time system variables

Table B-1 lists the date and time variables that are available in DB2 Web Query.

Table B-1 Date and time variables

System variable	Description	Format or value
&DATE	Returns the current date.	MM/DD/YY
&DMY	Returns the current date.	DDMMYY
&DMYY	Returns the current (four-digit year) date.	DDMMCCYY
&MDY	Returns the current date. Useful for numerical comparisons.	MMDDYY
&MDYY	Returns the current (four-digit year) date.	MMDDCCYY
&TOD	Returns the current time you entered DB2 Web Query.	HH.MM.SS
&YMD	Returns the current date.	YYMMDD
&YYMD	Returns the current (four-digit year) date.	CCYYMMDD

Date format

The various date formats enable you to define a field as a date and work with it as a date. Using the date format, you can perform the following tasks:

- ▶ Define date components, such as year, quarter, month, day, and day of week, and extract them easily from the date fields.
- ▶ Sort reports into date sequence, regardless of how the date appears. For example, January sorts before April even though, without date smarts, April alphabetically comes before January.
- ▶ Do arithmetic with dates and compare the dates without resorting to special date-handling functions.

Date format display options

The date format does not specify type or length. Instead, it specifies date component options (D, W, M, Q, Y, and YY) and display options. These options are shown in Table B-2.

Table B-2 Date format options

Display option	Meaning	Effect
D	Day	Displays a value from 1 to 31 for the day.
M	Month	Displays a value from 1 to 12 for the month.
Y	Year	Displays a two-digit year.
YY	Four-digit year	Displays a four-digit year.
T	Translate month	When used with M in a date (MT or TM), the three-letter abbreviation for the month in uppercase is displayed.

Display option	Meaning	Effect
t	Translate month	When used with M in a date (Mt or tM), the three-letter abbreviation for the month is displayed, capitalizing only the first letter of the month or day.
TR	Translate month or day	TR is like T, but displays the full name in uppercase.
tr	Translate month or day	tr is like t, but displays the full name in mixed case.
Q	Quarter	Displays the quarter Q1 - Q4.
W	Day-of-Week	On its own, W displays the number of the day of the week (1-7, Mon=1). Used in combination with other date options, W displays a three-letter abbreviation of the day of the week in uppercase.
w	Day-of-Week	Functions as uppercase W (described previously), except that the first letter is uppercase and the following letters are lowercase.
WR	Day-of-Week	Functions the same as uppercase W (described above), except that the entire day name is displayed instead of an abbreviation.
wr	Day-of-Week	Functions the same as lowercase w (described above), except that the entire day name is displayed instead of an abbreviation.
JUL	Julian format	Displays date in Julian format.
YYJUL	Julian format	Displays a Julian format date in the format YYYYDDD. The 7-digit format displays the four-digit year and the number of days counting from January 1. For example, January 3, 2001 in Julian format is 2001003.

Table B-3 shows samples of output for various date formatting options.

Table B-3 Sample output for date formatting options

Translation	Display
MT	JAN
Mt	Jan
MTR	JANUARY
Mtr	January
WR	MONDAY
wr	Monday
Q	Q1
YQ	07Q1

Controlling the date separator

You can control the date separators when the date is displayed. In basic date format, such as YMD and MDYY, the date components are displayed separated by a slash character (/). The same is true for the year-month format. The year-quarter format is displayed with the year and quarter separated by a blank (for example, 94 Q3 or Q3 1994). The single component formats display just the single number or name.

The separating character can be changed to a period, a dash, or a blank, or can be eliminated entirely. Table B-4 shows the FORMAT specifications that can be used to change the separating character.

Table B-4 Date separators

Format	Display
YMD	93/12/24
Y.M.D	93.12.24
Y-M	93-12
YBMBD	93 12 24 (The letter B signifies blank spaces.)
YIMID	931224 (The concatenation symbol "I" eliminates the separation character.)

Using date fields

Table B-5 shows valid examples of specifying dates.

Table B-5 Examples of specifying the dates

Situation	Natural date literal
In WHERE screening	WHERE MYDATE IS 'APR 25 1999'
In arithmetic expressions	MYDATE - '1999 APR 25'
In computational date comparisons	IF MYDATE GT '25 APR 1999'

Date fields in arithmetic expressions

The general rule for manipulating date fields in arithmetic expressions is that date fields in the same expression must specify the same date components. The date components can be specified in any order, and display options are ignored. Valid date components are Y or YY, Q, M, W, and D.

For example, NEWQUARTER and THISQUARTER both have FORMAT specifications of Q, and the value of THISQUARTER is 2. In this case, consider the following statement:

```
NEWQUARTER = THISQUARTER + 3
```

This statement gives NEWQUARTER a value of 1 (that is, the remainder of 5 divided by 4).

The following example calculates the number of days elapsed since January 1, 1999:

```
YEARTODATE = ORDERDATE - 'JAN 1 1999' ;
```

Converting date fields

You can also convert date fields. Two types of conversions are possible: format conversion and date component conversion. In *format conversion*, the value of a date format field can be assigned to an alphanumeric or integer field that uses date display options; the reverse conversion is also possible.

In *date component conversion*, a field whose format specifies one set of date components can be assigned to another field by specifying different date components. For example, the value of REPORTDATE (DMY) can be assigned to SALESDATE (Y); in this case, the year is extracted from REPORTDATE. If REPORTDATE is Apr 27 99, SALESDATE is 99.

Date functions

In this section, we present a few of the more common functions that you can do with dates. These functions are described in more detail in the help text included with Developer Workbench.

DATEADD: Adding or subtracting a date unit to or from a date

You can add or subtract years, months, days, weekdays, or business days from your date. Business days can take a holiday file as input. By default, a business day and a weekday are the same concept. In the following function, the date field must have a format like YYMD, MDY, or JUL. Increment must be an integer.

```
DATEADD(date, 'Y/M/D/WD/BD', increment)
```

Figure B-1 demonstrates adding 11 business days ('BD') to the order date to calculate the ship date.

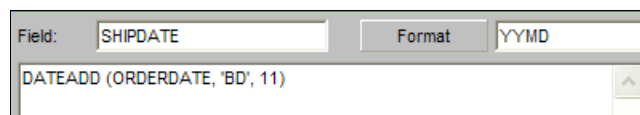


Figure B-1 DATEADD

The DATEADD function is not required if you simply want to add months or days to a date. To add months, your result field must be in a format similar to 'YYM' or 'MY'. To add days, your result field must contain days, for example, YYMD.

If we want to add 11 days to SHIPDATE in the previous example and do not need to worry about business or weekdays, we replace the DATEADD with the following values:

```
ORDERDATE + 11
```

This creates a SHIPDATE of 11 days in the future.

DATEDIFF: Calculating the difference between two dates

Similar to DATEADD, the DATEDIFF function allows you to find the difference between two dates in terms of years, months, days, weekdays, and business days. If you only want to find the difference in months or days, you do not need to use DATEDIFF. You can simply code a statement like the following example:

```
ORDERDATE - SHIPDATE
```

The resulting value is based on the least significant component in the order and ship dates. For example, if one of the dates is defined as YYM, then your results are in months. If both dates are defined down to the day level, then your results are in days.

The result field from DATEDIFF should be defined as an I8 field (Figure B-2):

```
DATEDIFF(date1, date2, 'Y/M/D/WD/BD')
```

Date1, date2, or both can be fields or constants, for example '20070101'.

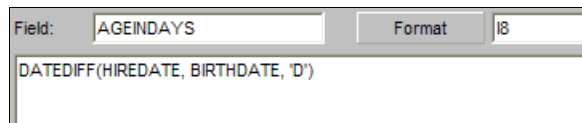
A screenshot of a database query editor. The 'Field:' label is followed by a text box containing 'AGEINDAYS'. To the right, a 'Format' label is followed by a dropdown menu showing 'I8'. Below these, a larger text box contains the SQL expression 'DATEDIFF(HIREDATE, BIRTHDATE, 'D')'.

Figure B-2 DATEDIFF

DATEMOV: Moving the date to a significant point

DATEMOV moves your date field to a significant point such as the end of the week or the beginning or the quarter. Table B-6 lists the possible values for move-point.

```
DATEMOV(date, 'move-point')
```

The date field must be a full date, for example, MDYY or YYJUL.

Table B-6 Values for move-point

Move-point	Meaning
EOM	End of month
BOM	Beginning of month
EOQ	End of quarter
BOQ	Beginning of quarter
EOY	End of year
BOY	Beginning of year
EOW	End of week
BOW	Beginning of week
NWD	Next weekday
NBD	Next business day
PWD	Prior weekday
PBD	Prior business day
WD-	Current weekday or prior weekday (if weekend)

Move-point	Meaning
BD-	Current business day or prior business day if current is a non-business day
WD+	Current weekday or next weekday (if weekend)
BD+	Current business day or next business day if current is a non-business day



Processing differences between the define and compute fields

In this appendix, we describe how DB2 Web Query processes a define or compute field at different points in the reporting sequence.

Figure C-1 shows the reporting sequence to illustrate the differences between the define and compute.

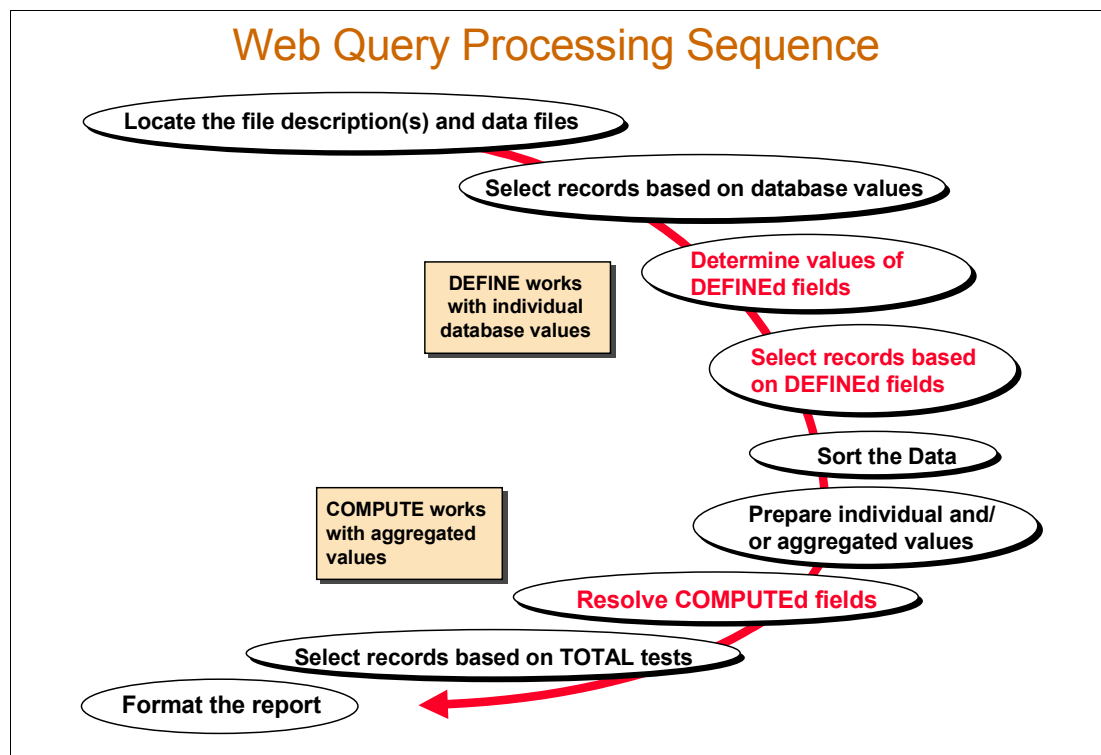


Figure C-1 DB2 Web Query processing sequence

We explain the sequence in greater detail as follows while noting the differences between the two fields:

1. When DB2 Web Query begins to process a report request, it locates the Master File Description (MFD) and the database. DB2 Web Query then processes the WHERE test on a database value to select a valid record for the report.
2. After selecting a record, DB2 Web Query determines the value of each define field, in sequence. After determining the values of the define fields, it executes any selection tests that use the value of a define field and potentially further limits the number of records per report.
3. DB2 Web Query then sorts the data (according to the BY and ACROSS phrases) and performs the action indicated in the verb phrase (PRINT, SUM, or COUNT). If the report uses SUM or COUNT, it aggregates the values in memory.
4. DB2 Web Query then takes the processed data and determines the values of any compute fields, in the sequence in which they appear in the request.
5. After evaluating the compute fields, DB2 Web Query performs the final selection test; it performs any WHERE TOTAL tests against any aggregated results and the values of the compute fields, and thereby further limits the records for the report.
6. Finally DB2 Web Query formats the report and routes it to your Web browser.



Setting up DB2 Web Query to run in an IASP environment

In this appendix, we explain how to set up IBM DB2 Web Query in an integrated auxiliary storage pool (IASP) environment, specifically:

- ▶ Storing the IBM DB2 Web Query integrated file system user objects in IASP directories, to use the environment on the second system when the IASP is allocated to that machine.
- ▶ Creating reports or graphs on data that is stored in the IASP database, to execute them alternatively from the system to which the IASP is allocated.

We assume that the switchable IASP environment is already configured and fully operational. It includes a “switchable” common IP interface that allows the user to use the same URL to connect to DB2 Web Query, regardless of the system that has allocated the IASP at a given moment.

In this appendix, *IASP520* is the IASP name and *IASP520DB* is the IASP database name. We refer to IBM DB2 Web Query as *QU2*.

On the first system

With the IASP device varied on, perform the following tasks:

1. Restore the CENTURY sample library in the IASP:

```
RSTLIB SAVLIB(CENTURY) DEV(*SAVF) SAVF(QGPL/CENTURY) RSTASPDEV(IASP520)
```

2. Install DB2 Web Query following the instructions explained in Chapter 2, “Installation and server operations” on page 17. Start it and verify that it is working correctly.

3. End DB2 Web Query.

4. Save the /qibm/UserData/webquery directory and subsequently move it in IASP, using a save file:

```
CRTSAVF FILE(QGPL/WEBQRYIFS) TEXT('WebQuery IFS USERDATA Objects')  
SAV DEV('/qsys.lib/qgp1.lib/WEBQRYIFS.file') OBJ('/qibm/UserData/webquery')  
DTACPR(*YES)
```

Note: Received i5/OS message 1450 objects saved in Beta3.

5. Allocate IASP to your job:

```
SETASPGRP ASPGRP(IASP520)
```

6. Verify that you can get to the sample data library, Century:

```
DSPLIB LIB(CENTURY)
```

7. Restore the /qibm/UserData/webquery directory in IASP:

```
RST DEV('/qsys.lib/qgp1.lib/WEBQRYIFS.file')  
OBJ('/qibm/UserData/webquery' *INCLUDE '/IASP520/webquery')
```

Note: Received i5/OS message 1450 objects saved in Beta3.

8. Rename /qibm/UserData/webquery directory to webqueryORI:

```
REN OBJ('/qibm/UserData/webquery') NEWOBJ(webqueryORI)
```

9. Add a symbolic link to map the newly restored IASP directory on the original path:

```
ADDLNK OBJ('/IASP520/webquery') NEWLNK('/qibm/UserData/webquery')  
LNKTYPE(*SYMBOLIC)
```

You see the message “Link added.”

10. Authorize user QTMHHTTP to the /qibm/UserData/WebQuery/ibi/webfocus76/* path:

```
CHGAUT OBJ('/qibm/UserData/WebQuery/ibi/webfocus76/*') USER(QTMHHTTP)  
DTAAUT(*RWX) OBJAUT(*ALL) SUBTREE(*ALL) SYMLNK(*YES)
```

11. Sign on as user QWEBQRYADM, allocate the IASP, and start DB2 Web Query:

```
SETASPGRP IASP520  
QWEBQRY76/STRWEBQRY
```

12. Configure the IASP database and metadata to the sample library tables in the browser interface of DB2 Web Query:

- a. Verify the IASP database name that is defined in the Relational Database Directory of your system:

WRKRDBDIRE

Figure D-1 shows the output of running this command. As you can see, in this example, the IASP DB name is set to IASP520DB. This is the name that we must reference when adding a new connection.

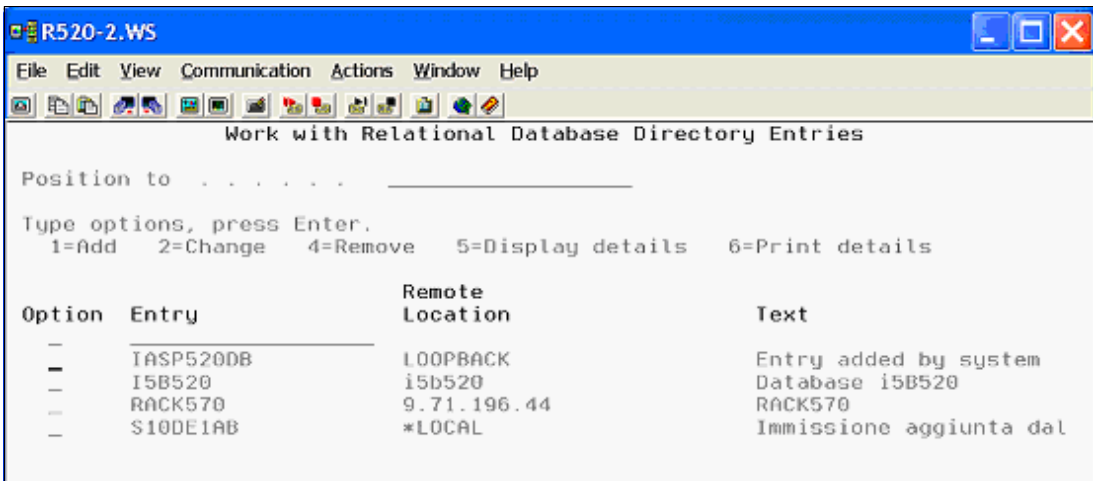


Figure D-1 WRKRDBDIRE display

- b. Log on to the browser interface. Select **Your_Domain** → **Reports**. Right-click **your_folder** and select **Metadata** as shown in Figure D-2.

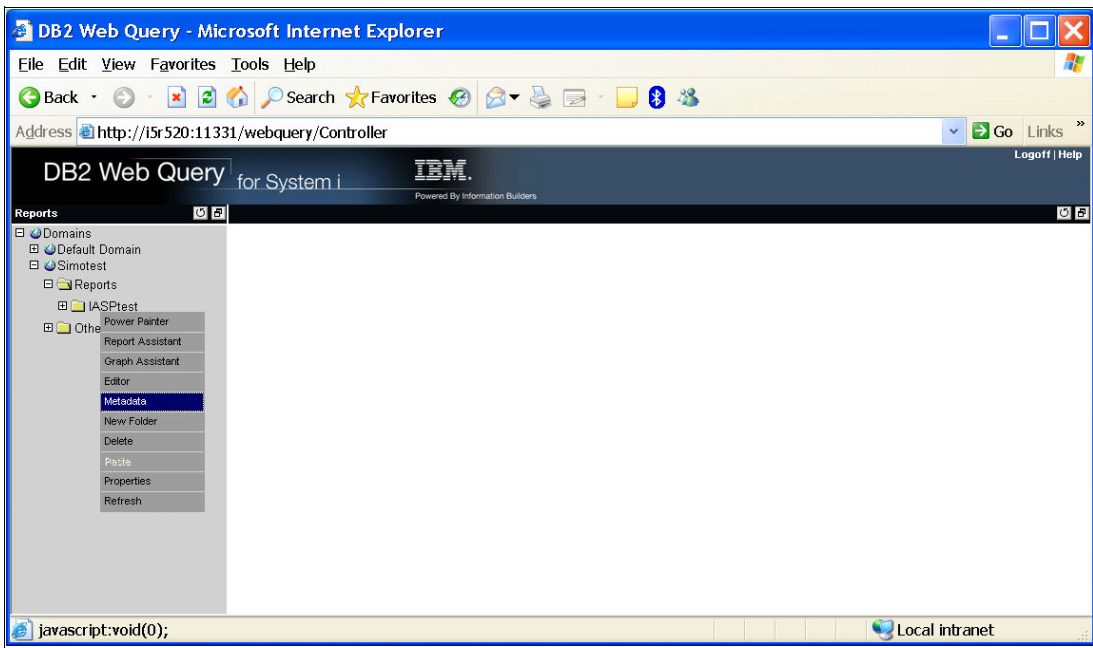


Figure D-2 DB2 Web Query metadata

- c. In the Data Adapters window (Figure D-3) that opens, define a new database connection allowing IBM DB2 Web Query to connect to the IASP Database instance, beside the SYSBASE (*LOCAL) database instance that already exists.
 - i. To define the new connection, in the left navigation pane, right-click **DB2 cli** and select **Add Connection**.
 - ii. In the right Add Connection for DB2 cli pane, enter a connection name (any name you want, the more representative the better) and the database instance name. In our example, we use IASP520 as the connection name and IASP520DB (as retrieved using the WRKRDBDIRE command) as the datasource name. Click **Configure** to confirm and add the connection.

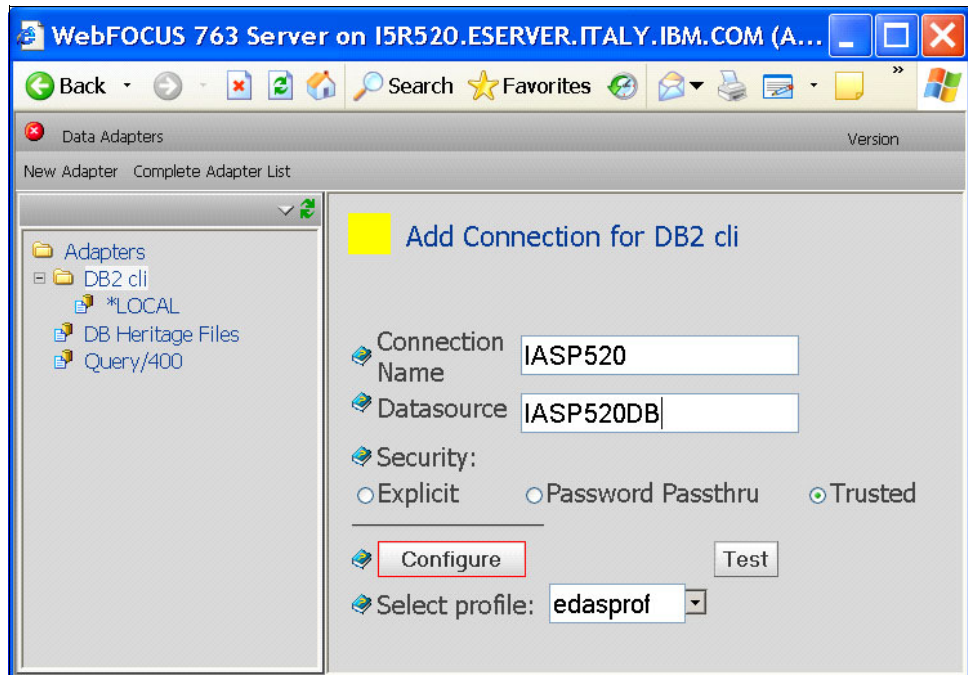


Figure D-3 Connection Name and Datasource name

13. Define a synonym to table CENTURY/ORDERS:
 - a. As shown in Figure D-4, in the left pane, right-click the newly defined connection, **IASP520**, and select **Create Synonym**.

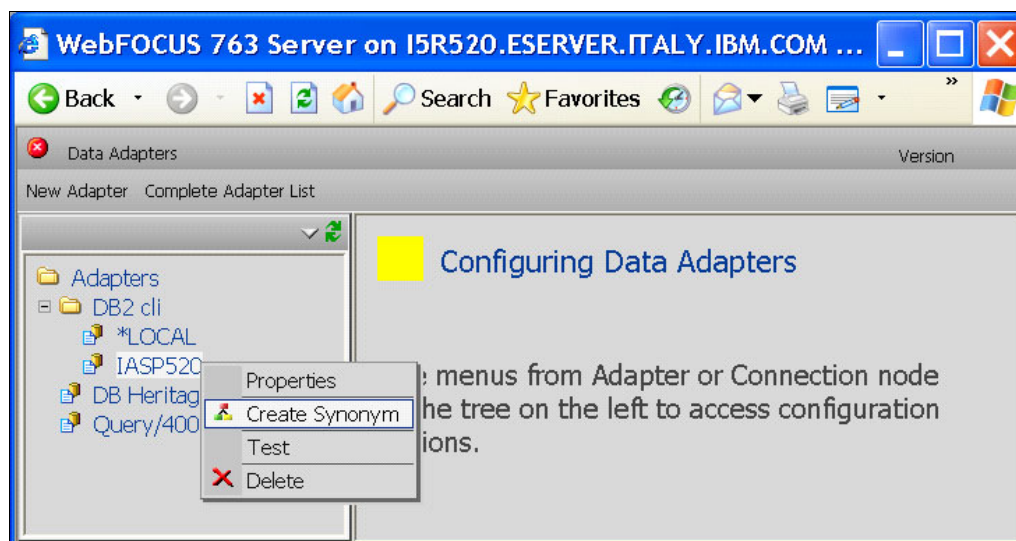


Figure D-4 Creating synonyms

- b. Create a report on Orders and test that everything works as expected.
Use Century, the sample data library, for your test, and add the Orders table, as you would do for any other library and table stored on the system.
 - c. Save your report.
14. Log off from the browser session.
15. Go back to the 5250 session, and sign on as QWEBQRYADM.
16. End DB2 Web Query by entering the following command:
QWEBQRY76/ENDWEBWRY

On the second system

In this section, we explain the steps that are required on the second system. It is important that we propagate the user profile QWEBQRYADM from the first system, using iSeries Navigator, to ensure that it has the same UID on both systems. Having a different UID can impact performance at IASP switch time.

1. On the first system, select **Users & Groups** → **Users**. Right-click **QWEBQRYADM** and select **Send**.

2. In the Send Users window (Figure D-5), add your second system. Under Available systems and groups, select **your system** and click **Add**. The system name moves to the Selected systems and groups pane. Click **OK**.

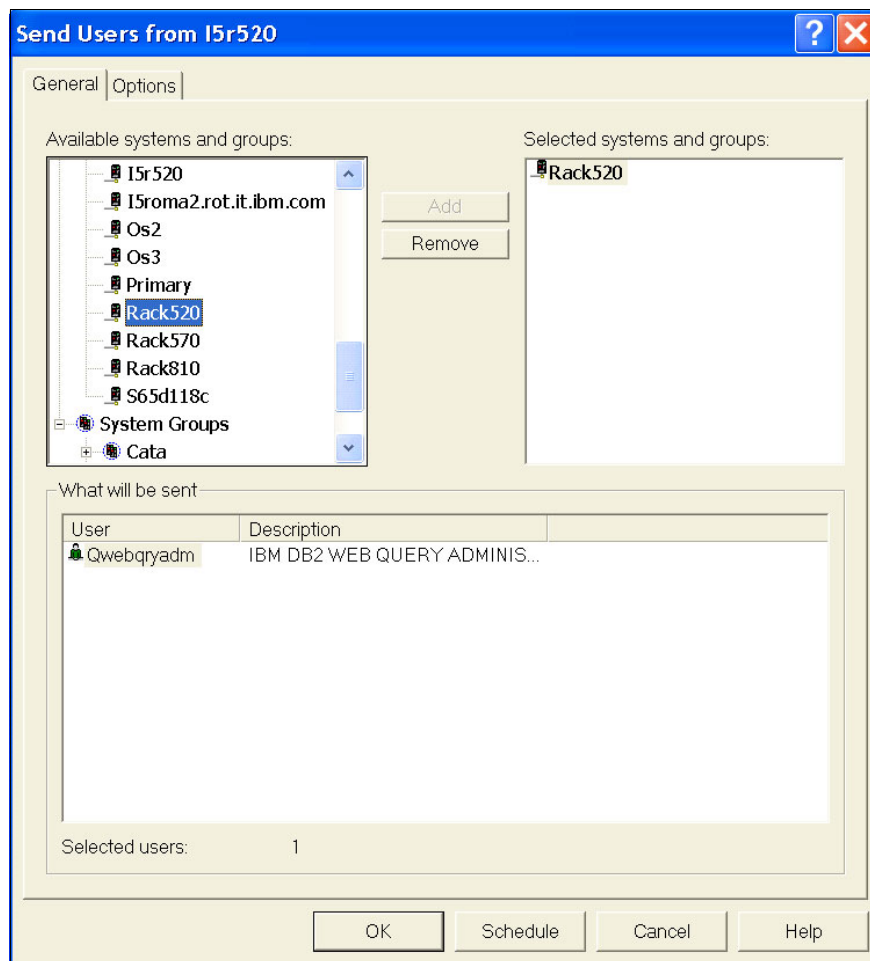


Figure D-5 Propagating users to a second system

3. Check that the two QWEBQRYADM user profiles have the same UID on both systems using iSeries Navigator.
 - a. On both systems, select **Users & Groups** → **Users**. Right-click **QWEBQRYADM** and select **Properties** → **Capabilities**.
 - b. In the Capabilities window (Figure D-6), click the **Unique Identifier** tab and check the UID. Click **OK**.

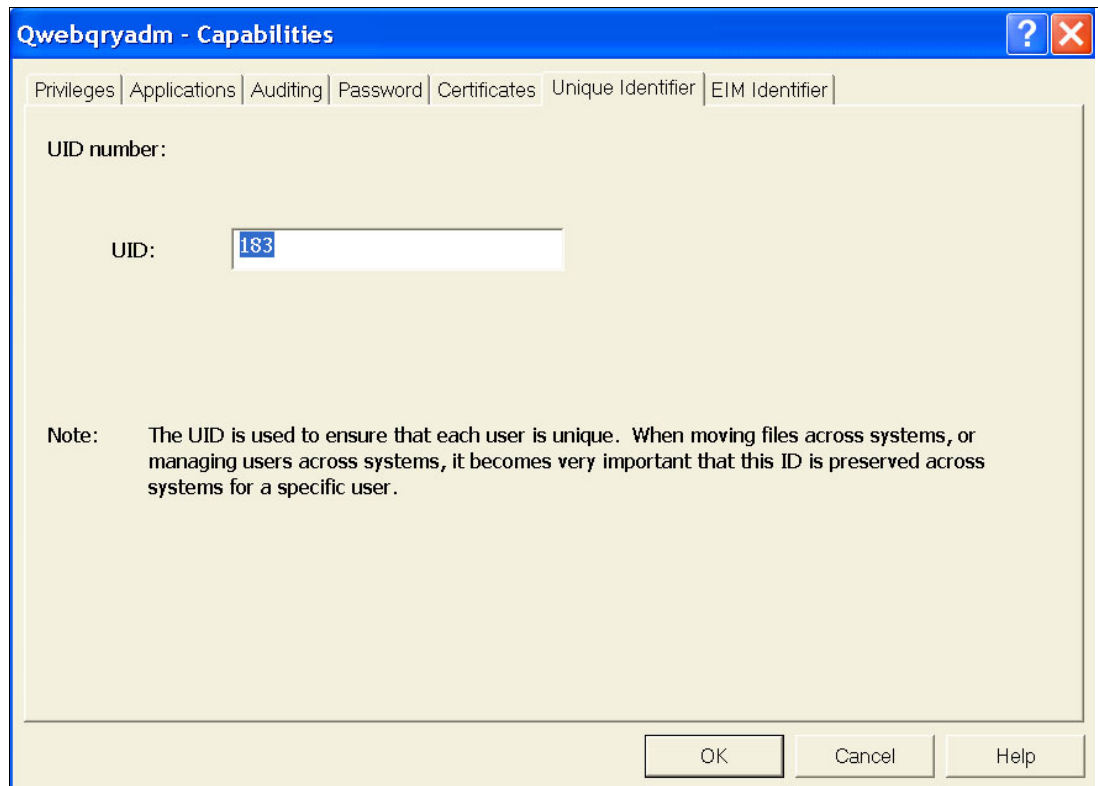


Figure D-6 Capabilities window showing the Unique Identifier tab

4. Install the DB2 Web Query as explained in Chapter 2, "Installation and server operations" on page 17.
5. Sign on as QWEBQRYADM and start DB2 Web Query.
6. Test DB2 Web Query to ensure that it is working correctly.
7. End DB2 Web Query.
8. Rename the /qibm/UserData/webquery directory to webqueryORI by entering the following command:


```
REN OBJ('/qibm/UserData/webquery') NEWOBJ(webqueryORI)
```
9. Switch IASP to a second system, using iSeries Navigator.
10. Allocate the IASP by typing the following command:


```
SETASPGRP ASPGRP(IASP520)
```

11. Create a symbolic link to directory /IASP520/webquery:

```
ADDLNK OBJ('/IASP520/webquery') NEWLNK('/qibm/UserData/webquery') LNKTYPE
(*SYMBOLIC)
```

12. Sign on as QWEBQRYADM.

13. Enter the SETASPGRP command.

14. Start DB2 Web Query.

15. Verify that you receive the reports or graphs that were created on the first system and that they are working correctly.



Additional material

This book refers to additional material that can be downloaded from the Internet as described below.

Locating the Web material

The Web material associated with this book is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser at:

<ftp://www.redbooks.ibm.com/redbooks/SG247214>

Alternatively, you can go to the IBM Redbooks Web site at:

ibm.com/redbooks

Select the **Additional materials** and open the directory that corresponds with the IBM Redbook form number, SG247214.

Using the Web material

The additional Web material that accompanies this book includes the following files:

<i>File name</i>	<i>Description</i>
QWQCENT.zip	iSeries Library

System requirements for downloading the Web material

The following list contains the most important requirements:

- ▶ i5/OS V5R4
- ▶ 5733-QU2 - DB2 Web Query for System i

How to use the Web material

Create a subdirectory (folder) on your workstation, and unzip the contents of the Web material zip file into this folder and upload the i5/OS save file to your System i environment.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

For information about ordering these publications, see “How to get IBM Redbooks” on page 387. Note that some of the documents referenced here might be available in softcopy only.

- ▶ *OnDemand SQL Performance Analysis Simplified on DB2 for i5/OS in V5R4*, SG24-7326
- ▶ *Preparing for and Tuning the SQL Query Engine on DB2 for i5/OS*, SG24-6598
- ▶ *SQL Performance Diagnosis on IBM DB2 Universal Database for iSeries*, SG24-6654

Online resources

These Web sites are also relevant as further information sources:

- ▶ Information Builders Corporate
<http://www.informationbuilders.com/>
- ▶ DB2 for i5/OS Portal
<http://www.ibm.com/systems/i/software/db2>
- ▶ DB2 Web Query
<http://www-03.ibm.com/systems/i/software/db2/webquery/>
- ▶ *System i Database DB2 UDB SQL call level interface (ODBC)*
<http://publib.boulder.ibm.com/infocenter/iseries/v5r4/topic/cli/rzadp.pdf>
- ▶ *Indexing and Statistics Strategies for DB2 for i5/OS*
<http://www.ibm.com/servers/enable/site/education/ibo/record.html?indxng>

How to get IBM Redbooks

You can search for, view, or download Redbooks, IBM Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

Symbols

*QRYDFN 4

Numerics

3020 record 327
5722-QU1 (Query/400) 4–5, 18
5733-QU2 (DB2 Web Query) 5–7, 17–18, 32, 173, 175, 324

A

access file 198, 366
Active Reports 8, 175, 343
 active visualizations 358
 ActiveX controls 181
 cross-tab or pivot functionality 189
 e-mail user requirements 359
 file size 358
 filtered chart 186
 general use of 181
 highlights 176
 license key 177
 overview 176
 problems saving 359
 report creation 177
 tutorial 175
Active Reports Styling 178
ActiveX controls 181
adapter 13
 optimization 312
 processing 312
add-on products 358
ADDRDBDIRE command 47
Administration Console 12
administrator 25
administrator tasks 28
 assignment of authority to users 31
 domains 28
 subfolder creation 30
aggregate field 90
aggregation optimization 319
AJAX 6, 343
analysis 328
application server 10–11
architecture 10
auditing 346
authority, assignment to users 31
authorization of users 19
automatic generation or refresh of metadata 347
autostart after IPL 345
Available fields icons 66

B

bar chart 110, 245
 dual axis 118
 multiple bars to one chart 114
 with drill-down capabilities 112
basic synonyms 266
basic user 25, 53
benchmark, DB2 Web Query 335
Business View 283

C

calendar controls 259
call-level interface (CLI) 23
capabilities of DB2 Web Query 342
CENTURY 32
Change report 31
chart, revenue trend over time 125
CHGUSRPRF 28
child table 39
Classic Query Engine (CQE) 5, 13, 61, 325, 350
CLI (call-level interface) 23
cluster 65
cluster join 275
Common Domain 28, 59
compound parameterized report 234
 bar chart 245
 gross profit report 246
 HTML page options 246
 HTML report 239
 line chart 241
 New form element 243
compute field 146, 375
computing fields 81
conditional styling 84, 146
configuration of Developer Workbench 194
Contains operand 185
CPU resource usage 337
CQE (Classic Query Engine) 5, 13, 61, 325, 350
CREATE ALIAS 38
Create domain 31
Create report 31
Create subfolder 31
Create Synonym 36
cross-system join 45, 322
cross-tab functionality 189
cross-tab report 79

D

dashboard 15, 354
 graphics 354
 periodic report 354
data adapters 7, 13
 DB2 CLI 7, 13

- DB2 Heritage File 7, 13
- DB2 Heritage File adapter 13
- Query/400 7
- data download to PC server or Windows server 344
- data field, format conversion 371
- data profiling 8, 288
- database
 - analysis and tuning 333
 - design 325
 - monitor 334
- date
 - component conversion 371
 - functions 371
 - manipulation 271
 - separator 370
- date and time
 - functionality 367
 - system variables 368
- Date Decomposition 8, 269
- date field 370
 - conversion 371
 - date component conversion 371
 - in arithmetic expressions 370
- date format 368
 - display options 368
- date formatting 75
- date range filter 133
- date range parameter 138
- DATEADD function 371
- DATEDIFF function 372
- DATEMOV function 372
- DB2 10
- DB2 CLI adapter 7, 13
 - performance 310
- DB2 file output 96
- DB2 for i5/OS
 - database, query without multiple licenses 349
 - optimization 325
 - referential integrity features 347
- DB2 for Linux, UNIX, or Windows 45
- DB2 Heritage File adapter 7, 13
- DB2 Web Query 5
 - additional features 343
 - administrator tasks 28
 - adopted authority for reports 346
 - aggregation optimization 319
 - architecture 10
 - attributes 11
 - autostart after IPL 345
 - base features 6
 - capabilities 342
 - conditions that prevent optimization 317
 - data adapters 13
 - Developer Workbench 14
 - domain authority 346
 - domains 28, 345
 - exception join 350
 - features 6
 - functional differences with Query/400 303
 - functional similarities with Query/400 302
 - general announcement 342
 - Graph Assistant 6
 - imported Query/400 object 351
 - imported Query/400 query 351
 - installation 18
 - installation and server operations 17
 - integration to portlets 344
 - integration with Excel 2007 358
 - integration with Rational tools 344
 - introduction 5
 - license keys 20
 - login 26
 - multiple licenses 349
 - named user 26, 346
 - object import from Query/400 350
 - OLAP feature dimensions 358
 - optional features from IBM 7
 - optional features from Information Builders 8
 - PC and System i requirements 21
 - performance benchmark 335
 - performance considerations 309
 - port 11331 for HTTP requests 344
 - Portal 6.0 344
 - Power Painter 6
 - product architecture 3
 - product information 342
 - product positioning 14
 - QRYDFN edit 307
 - Query/400 definitions 351
 - reasons to move from Query/400 342
 - referential integrity 39
 - Report Assistant 6
 - report auditing 346
 - report development 351
 - report run from CL program 354
 - Reporting Server 12
 - commands 23
 - reports from Query/400 350
 - running existing Query/400 reports 301
 - running in an IASP environment 377
 - RUNWEBQRY command 354
 - save and restore 360
 - server jobs 23
 - server-based processing 345
 - servlet 11
 - sign-on via LDAP or Kerberos 346
 - sorting optimization 319
 - stored procedures with input parameters 353
 - training and education 362
 - upgrade to 350
 - user security 345
 - web console 344
- define field 146, 375
- defining fields 81
- detail reports 68
- developer 25
- developer tasks 32
 - creation of metadata 34
 - cross-system join 45
- Developer Workbench 7, 14, 63, 193, 343

- additional features 265
- Business View 283
- configuration 194
- data profiling 288
- Date Decomposition 269
- date manipulation 271
- default options 198
- filter creation 273
- HTML Layout Painter 233
- Impact Analysis 285
- installation 19
- Java 2 SDK 23
- not part of base DB2 Web Query product 358
- OLAP 201
- profiling 193
- requirements 22
 - hardware 22
 - software 22
- SQL Wizard 291
- Synonym Editor 193, 266
- synonym management 348
- tutorial 193
- dicing 7
- dimensions 203
- domain 27–28, 345
 - authority 346
 - create 29
 - creation 29
 - eight-character rule 29
 - rename 30
 - renaming 30
- domain-level report movement 348
- double-spaced printing 354
- drill down 15, 117
 - adding to chart 116
 - bar chart 112
- drill-down report 102
 - further drill downs 354
- DSPFFD 34

E

- EDAPGWY 23
- EDAPLOG 23
- EDAPTH 23
- ENDWEBQRY command 23
- enterprise resource planning (ERP) 8
- ERP (enterprise resource planning) 8
- Excel 6
 - output 95
 - with formulas option 358
- Excel 2000 95
- Excel 2007 integration with DB2 Web Query 358
- Excel 97 95
- Excel spreadsheet
 - access to another spreadsheet 358
 - integration 356
 - report 356
- exception joins 350
- Extensible Markup Language (XML) 6

F

- FAQs (Frequently Asked Questions) 341
- federated DB2 7
- filter creation 273
- Foreign key option 66
- foreign key relationships 320
- format conversion 371
- Frequently Asked Questions (FAQs) 341
- fullselect 320

G

- general announcement 342
- generic term report 234
- graph
 - building with Power Painter 157
 - properties and settings, Graph Assistant 124
- Graph Assistant 6
 - bar chart 110
 - conditional styling 146
 - date range parameter 138
 - drill down 116
 - dual axis bar and line chart 118
 - graph properties 124
 - line graph 126
 - pie charts 140
 - traffic lighting a graph 146
 - tutorial 109
- gross profit report 246
- group profile MDUNTITLED 58
- GRPPRG user profile, domain authority 346

H

- hierarchy 203
- Highlight button 185
- highly parameterized report 248
 - calendar controls 259
- HLISNK 23
- host table 317
- Href 28
- HTML Layout Painter 193, 233
 - compound parameterized report 234
 - highly parameterized report creation 251
 - parameter controls 256
 - Run button 247
 - tool 8
 - tutorial 233
- HTML page options 246
- HTML report 239
- HTTP clients 10

I

- i5/OS system objects 19
- IBM DB2 Web Query for System i 5
- IBM Query for System i 4
- icons for Available fields 66
- IDE (Integrated Development Environment) 14
- Impact Analysis 8, 285
- imported report 351

- index 320, 326
- Index Advisor 327, 334
- Index Only Access 326, 334
- Information Builders optional features 8
- Inner Join 107
- input parameters 353
- installation 17–18
- Integrated Development Environment (IDE) 14
- integrated file system 11
 - directories 19
 - metadata 365
- integration with Rational tools 344
- iSeries Navigator, synonym management 347

J

- Java 2 SDK 1.5.0_09 23
- join disablers 317
- join optimization 316
- join relationship 347
- join types 107
- join, table 275
- joining tables 106
- JSCOM3 23

K

- Kerberos sign-on 346
- key constraints 41
- key performance indicators (KPI) 15

L

- LDAP sign-on 346
- Left Outer Join 107
- library list-like behavior 348
- license key 20, 177
- line chart 118, 241
- line graph 126
 - date range filter 133
- long-running report 328

M

- Managed Reporting drivers 12
- master file 198, 365
- materialized query table (MQT) 37, 321, 334
- MDUNTITLED group profile 19, 58
- memory usage 338
- metadata 5, 34
 - aliases 37
 - automatic generation or refresh 347
 - considerations for creation 37
 - deletion of 366
 - from Query/400 objects 304
 - MQT 37
 - prefix and suffix 307
 - referential integrity 38
 - stored procedures 37
 - tables 37
 - views 37
- metadata creation

- i5/OS object change 38
- i5/OS object deleted 38
- miscellaneous considerations 37
- multimember files 37
 - on tables with referential integrity 43
 - stored procedure 37
- model 8nn system 344
- MQT (materialized query table) 321, 334
- MRADMIN group 58
- MRADMIN group profile 19
- multi-dimensional 7
- multimember files 37
- multiple bars on one chart 114
- multiple instance 107
- multiple members 13
- multiple result sets 37
- multiplicative effect 317

N

- named user 26, 346
 - licenses 8
- New form element 243
- number of rows 352

O

- offline reporting 8
- OLAP
 - dicing 212
 - dimension 202
 - Dimension Builder 205
 - dimensions for database 358
 - drill down 202
 - drill-down options 211
 - enabled report 212
 - enablement 210
 - getting started 202
 - grouping data 231
 - hierarchy 202–203
 - hyperlinks 202
 - measure 202
 - metadata 203
 - modeling view 204
 - pivot 202
 - slicing 212
 - Synonym menu 205
 - terminology 202
 - tutorial 201
- OLAP (online analytical processing) 7, 201
- OLAP Control Panel 224
- OLAP module 343
- OLAP report
 - on demand column sort 358
 - prior selections 358
- on-demand paging 93
- online analytical processing (OLAP) 7, 201
- operational reports 15
- OPNQRYF command 13
- optimization 317
 - DB2 for i5/OS 325

- disablers 328
- hierarchy 315
- other influencers 319
- Other Files folder 28
- output files for next query 303
- output options 92

P

- palettes 154
- parameter controls 256
- parameter passing 303
- parameterized report 98
 - design 356
 - multiple item selection 353
- parameters, adding 234
- parent report 102
- parent tables 39
- PC requirements 21
- PC server 344
- PDF (Portable Document Format) 6
 - distributing batch reports via e-mail 359
 - output 94
- performance
 - adapter processing and optimization 312
 - basics 310
 - benchmark for DB2 Web Query 335
 - case study 327
 - DB2 CLI adapter 310
 - DB2 Web Query 309
 - Query/400 adapter 324
 - remote database access 322
- pie chart 140
 - product type for each year 140
- pivot functionality 189
- port 11331 for HTTP requests 344
- Portable Document Format (PDF) 6
- Portal 6.0 integration to portlets 344
- portlet integration with Portal 6.0 344
- PostScript 6
- Power Painter 6
 - building a graph 157
 - building a report 165
 - Clustered Bar chart 158
 - dashboard 354
 - Data Sources palette 156
 - launching 154
 - PDF report 172
 - Properties palette 156
 - tailoring the main window 154
 - tutorial 153
- prefix 36
 - and suffix 36
 - metadata 307
- primary key 39
- printing, double-spaced 354
- product architecture 3
- product information 342
- projection 315

Q

- QP0ZSPWP 24
- QRYDFN
 - edit by DB2 Web Query 307
 - objects 301
- QSQSRVR 24
- QSYSWRK subsystem 24
- Query/400 4, 301–302
 - 5722-QU1 5
 - adapter 7, 324
 - adapter performance 324
 - definitions 350–351
 - functional differences with DB2 Web Query 303
 - functional similarities with DB2 Web Query 302
 - imported query 351
 - metadata from Query/400 objects 304
 - migration of reports to DB2 Web Query 350
 - object 351
 - object import 350
 - product history 4
 - reasons to move to DB2 Web Query 342
 - report development 351
 - running existing reports in DB2 Web Query 301
- Quick Data 9
- QWEBQRYADM 19
- QWQCEN 32

R

- ranking columns 91
- Rational tools 344
- RDBMS (relational database management system) 10
- Redbooks Web site 387
 - Contact us xiv
- referential constraint 39
- referential integrity 38, 43, 316
 - benefits in DB2 Web Query 39
 - DB2 for i5/OS 347
 - metadata 38
 - metadata creation on tables 43
- Refresh Synonyms 348
- registered users 26
- registration
 - named users 26
 - new user 26
 - removal 26
- relational database management system (RDBMS) 10
- remote database access 322
- report
 - adopted authority 346
 - building with Power Painter 165
 - creating with Active Reports 177
 - creation 64
 - development 351
 - development phase 352
 - distributing batch reports in PDF via e-mail 359
 - execution phase 311
 - layout 65
 - long-running 328
 - production phase 312

- Query/400 reports in DB2 Web Query 301
- request process flow 310
- sample detail 80
- security for specific user 346
- selection defined on expression 356
- tuning options 330
- types 15, 98
- Report Assistant 6
 - date formatting 75
 - DB2 file output 96
 - drill-down reports 102
 - Excel output 95
 - footings 86
 - generating subtotals 88
 - headings 86
 - on-demand paging 93
 - output options 92
 - parameterized reports 98
 - PDF output 94
 - ranking columns 91
 - report properties 105
 - selection criteria 76
 - sort by aggregate field 90
 - sorting 90
 - style sheet 351
 - subtotal generation 88
 - tutorial 63
- report developer, SQL statements 359
- Report Library 15
- Reporting Server 10, 12
 - commands 23
- Reports folder 28
- requirements 21
- restore 360
- revenue trend over time chart 125
- Right Join 302
- row quantity 352
- RSTLCPGM command 17
- Run button execution 247
- Run report 31
- RUNWEBQRY command 354
 - database file option 355
 - registered named user 356

S

- sample detail report 80
- save 360
- scorecards 15
- second Y axis 121
- security
 - specific user 346
 - user 345
- selection 314
- selection criteria 76
- server jobs 23
 - EDAPGWY 23
 - EDAPLOG 23
 - EDAPTH 23
 - HLISNK 23
 - JSCOM3 23

- QP0ZSPWP 24
- QSQSRVR 24
- TSCOM3 23
- WQLWI7 24
- server operations 17
- server-based processing 345
- servlet container 11
- servlet engines 11
- single instance 107
- slicing 7
- sort by aggregate field 90
- sorting optimization 319
- spreadsheet reports in Excel 356
- SQE (SQL Query Engine) 5, 350
- SQE Plan Cache 327, 333
- SQL catalog views 61
- SQL CREATE TABLE 61
- SQL fullselect 320
- SQL Performance Monitor 327
- SQL Plan Cache snapshot 327
- SQL Query Engine (SQE) 5, 350
- SQL statements, report developer 359
- SQL trace 328
- SQL translation 315
- SQL view 330, 353
 - creation 319
 - new report 332
- SQL Wizard 8, 193, 291
- statistics 326
- stored procedure
 - input parameters 353
 - metadata creation 37
- STRWEBQRY command 23
- style sheet 351
- subfolders creation 30
- suffix 36
 - and prefix 36
 - metadata 307
- sum reports 68
- summary report 65
- SUPGRPPRF parameter 346
- Supplemental Group (SUPGRPPRF) parameter 346
- synonym 5, 34, 330
 - individual movement 361
 - management 347
- Synonym Editor 193, 265–266, 348
- Synonyms, Refresh option 348
- System i
 - objects 61
 - requirements 22
- system objects 19

T

- table join 275
- target table 317
- time and date functionality 367
- trace statements 328
- traffic lighting
 - a graph 146
 - conditional styling 84

- TRCWEBQRY command 23
- TSCOM3 23
- tutorial
 - Active Reports 175
 - Developer Workbench 193
 - getting started 57
 - Graph Assistant 109
 - HTML Layout Painter 233
 - OLAP 201
 - Power Painter 153
 - Report Assistant 63

U

- unique join 317
- upgrade to DB2 Web Query 350
- user authorization and verification 19
- user security 345
- users
 - displaying 26
 - registration 26
 - removing 26
- user-specified date range parameter 138

V

- variables in report headings and footings 86
- verification of users 19
- view 320
- Visual Explain 327, 334

W

- WAB (Web Application Bundle) file 12
- Web 2.0 343
- Web Application Bundle (WAB) file 12
- Web browser clients 11
- web console 344
- Web server 10–11
- Web-based query and report writing function 343
- WebFOCUS 235
 - reports 9
- WebFocus ReportCaster 9
- WebSphere Application Server 345
- WHERE 77
- WHERETOTAL 77
- Windows Explorer, synonym management 347
- Windows server 344
- WQLWI7 24
- WRKLICINF command 20
- WRKLNK command, synonym management 347
- WRKRDBDIRE command 46

X

- XML (Extensible Markup Language) 6

Y

- Y axis 121



Getting Started with DB2 Web Query for System i

(0.5" spine)
0.475" <-> 0.873"
250 <-> 459 pages



Getting Started with DB2 Web Query for System i



Redbooks

Learn DB2 Web Query by using the easy to follow tutorials

Follow the installation and best practices guides

Take your Query/400 reports to the next level

The DB2 Web Query for System i product is a Web-based query and report writing product that offers enhanced capabilities over the IBM Query for iSeries (also commonly known as Query/400) product. IBM DB2 Web Query for System i includes Query for iSeries technology to assist customers in their transition to DB2 Web Query. It offers a more modernized, Java-based solution for a more robust, extensible, and productive reporting solution. DB2 Web Query provides Report Assistant, Graph Assistant, and Power Painter. Using these tools, customers can modernize existing Query for iSeries reports while providing a foundation for building more complex business intelligence applications, such as online analytical processing (OLAP), data mining, dash boarding, or data warehouse implementations.

This IBM Redbook publication gives a broad understanding of the new DB2 Web Query product. It entails a group of self-explanatory tutorials to help you get up to speed quickly. Overall this book is designed for IT users. You can extract and use Part 2, "Tutorials for DB2 Web Query," and Part 3, "Tutorials for DB2 Web Query optional features," as stand-alone tutorials for anyone who is developing their own queries. Much of Chapter 8, "Active Reports," is appropriate for those who work disconnected from the System i environment. In addition, you can use much of Chapter 10, "OLAP (Online Analytical Processing)," if you will simply be running OLAP-enabled reports.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-7214-00

ISBN 0738486760