Redbooks
ibm.com/redbooks

IBM DB2 Web Query for i

The Nuts and Bolts

Hernando Bedoya

Jim Bainbridge

Rob Bestgen

Mike Cain

Doug Mack

Simona Pacchiarini

Kathryn Steinbrink

Lin Lin Su

Kevin Trisko

Analytics

Power Systems

IBM

Redbooks

International Technical Support Organization

IBM DB2 Web Query for i: The Nuts and Bolts

April 2017

SG24-8379-00

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

First Edition (April 2017)

This edition applies to Version 2.2 of DB2 Web Query for i and IBM i 7.1, 7.2 and 7.3.

Contents

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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 grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US

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 jurisdictions 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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

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

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

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.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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 actual people or business enterprises 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. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

DB2®
IBM®
Power Systems™
Redbooks®
Redbooks (logo) ®
System i®
z/OS®

The following terms are trademarks of other companies:

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

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

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

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

Preface

Business Intelligence (BI) is a broad term that relates to applications that analyze data to understand and act on the key metrics that drive profitability in an enterprise. Key to analyzing that data is providing fast, easy access to it while delivering it in formats or tools that best fit the needs of the user.

At the core of any BI solution are user query and reporting tools that provide intuitive access to data supporting a spectrum of users from executives to “power users,” from spreadsheet aficionados to the external Internet consumer.

IBM® DB2® Web Query for i offers a set of modernized tools for a more robust, extensible, and productive reporting solution than the popular IBM Query for System i® tool (also known as IBM Query/400). IBM DB2 Web Query preserves investments in the reports that are developed with Query/400 by offering a choice of importing definitions into the new technology or continuing to run existing Query/400 reports as is. But, it also offers significant productivity and performance enhancements by leveraging the latest in DB2 for i query optimization technology.

The DB2 Web Query product is a web-based query and report writing product that offers enhanced capabilities over the IBM Query for iSeries product (also commonly known as Query/400). IBM DB2 Web Query for 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 the ability to query or build reports against data that is stored in DB2 for i (as well as other DB2 and non DB2) databases through browser-based user interface technologies:

- Build reports with ease through the web-based, ribbon-like InfoAssist tool that leverages a common look and feel that can extend the number of personnel that can generate their own reports.
- Simplify the management of reports by significantly reducing the number of report definitions that are required through the use of parameter driven reports.
- Deliver data to users in many different formats, including directly into spreadsheets, or in boardroom-quality PDF format, or viewed from the browser in HTML.
- Leverage advanced reporting functions, such as matrix reporting, ranking, color coding, drill-down, and font customization to enhance the visualization of DB2 data.

DB2 Web Query offers features to import Query/400 definitions and enhance their look and functions. By using it, you can add OLAP-like slicing and dicing to the reports or view reports in disconnected mode for users on the go.

This IBM Redbooks® publication provides a broad understanding of what can be done with the DB2 Web Query product.

Authors

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 a Senior IT Specialist at STG Lab Services and Training in Rochester, Minnesota. He writes extensively and teaches IBM classes worldwide in all areas of DB2 for i. Before joining STG Lab Services, he worked in the ITSO for nine years writing multiple IBM Redbooks publications. He also worked for IBM Colombia as an IBM AS/400 IT Specialist doing presales support for the Andean countries. He has 28 years of experience in the computing field and has taught database classes at Colombian universities. He holds a master's degree in Computer Science from EAFIT, Colombia. His areas of expertise are database technology, performance, and data warehousing. Hernando can be contacted at hbedoya@us.ibm.com.

Jim Bainbridge is a senior DB2 consultant on the DB2 for i Center of Excellence team in the IBM Lab Services and Training organization. His primary role is training and implementation services for IBM DB2 Web Query for i and business analytics. Jim began his career with IBM 30 years ago in the IBM Rochester Development Lab, where he developed cooperative processing products that paired IBM PCs with IBM S/36 and AS/400 systems. In the years since, Jim has held numerous technical roles, including independent software vendors technical support on a broad range of IBM technologies and products, and supporting customers in the IBM Executive Briefing Center and IBM Project Office.

Rob Bestgen is a member of the DB2 for i Center of Excellence team helping customers use the capabilities of DB2 for i. In addition, Rob is the chief architect of the DB2 SQL Query Engine (SQE) for DB2 for i and is the product development manager for DB2 Web Query for i.

Mike Cain is a Senior Technical Staff Member within the IBM Systems and Technology Group. He is also the founder and team leader of the DB2 for i Center of Excellence in Rochester, Minnesota, US. Before his current position, he worked as an IBM AS/400 Systems Engineer and technical consultant. Before joining IBM in 1988, Mike worked as a System/38 programmer and data processing manager for a property and casualty insurance company. Mike has 26 years of experience with IBM, engaging clients and IBM Business Partners around the world. In addition to assisting clients, he uses his knowledge and experience to influence the IBM solution, development, and support processes.

Doug Mack is a DB2 for i and Business Intelligence Consultant in the IBM Power Systems™ Lab Services organization. Doug's 30-plus year career with IBM spans many roles, including product development, technical sales support, Business Intelligence Sales Specialist, and DB2 for i Product Marketing Manager. Doug is a featured speaker at User Group conferences and meetings, IBM Technical Conferences, and Executive Briefings.

Simona Pacchiarini has worked for IBM Italy since 1989. Over the years, she has participated in IBM Redbooks publication projects on client connectivity, Windows cooperation, and database, first on the IBM AS/400 and now on IBM i. She is working in STG LAb Services in Italy, assisting customers on DB2 Web Query for i projects, DB2 for i performance, and IBM i system performance studies.

Kathryn Steinbrink is an Advisory Software Engineer for IBM i Development. She joined IBM in 1979. She has worked in a variety of functional areas, but the last half of her career has been in the area of DB2 for i. She is the development team leader for DB2 Web Query. She can be reached at krs@us.ibm.com.

Lin Lin Su is a Software Engineer in IBM China Systems & Technology Lab. She joined IBM in 2016 and now is responsible for DB2 Web Query for i testing. She has been working for DB2 Web Query for i 2.1, 2.1.1 and 2.2 Hotfix testing. She can be reached at slbj@cn.ibm.com.

Kevin Trisko is a member of the IBM i Global Support Center where his focus is with the Database and DB2 Web Query for i teams. He's worked on the IBM i platform since 1990 in various roles - Testing, Database Development, Database Administration/Engineer, and Support Center.

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

Scott Vetter and Debra Landon
International Technical Support Organization, Rochester Center

Cindy Mestad
IBM STG Lab Services Rochester

Kathryn Steinbrink
IBM Development Rochester

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

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 books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

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

  ibm.com/redbooks

- Send your comments in an email 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

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

1

The cycle of Business Intelligence and IBM DB2 Web Query for i

Analytics is a broad term that encompasses many different types of analytical solutions. Business Intelligence (BI) is an established term to describe elements that now fall into the Analytics scope. BI is generally associated with reporting and data visualization solutions and infrastructures to support that, such as a data warehouse or operational data store (ODS), which is described later in this chapter.

IBM DB2 Web Query for i is a BI tool that is designed for the IBM i environment. DB2 Web Query DataMigrator ETL Extension is a member of the DB2 Web Query for i family that can be leveraged to automate the consolidation, movement, and transformation of data to prepare it for analytics. Both of these products can use the DB2 for i capabilities to support analytical applications. With DB2 Web Query and DB2 for i, you can set a foundation for analytics that provides the BI platform and can be the engine to feed into other, advanced analytics toolsets.

This chapter is intended to lay out the process of implementing a BI solution by using DB2 Web Query and DB2 for i.

1.1 The process of implementing Business Intelligence with DB2 Web Query for i

Figure 1-1 shows the process cycle, which is a cycle because it is repeated as your analysis focus expands. The process is a general set of preferred practices and set of considerations that should not look that different from other application development processes. But it is a reminder that a BI application implementation should not shortcut normal preferred practices and procedures. Unfortunately, in many cases shortcuts or approaches to reporting solutions can often lead to a “do-over” later. Examples include the “if you build it they will come” approach where IT tries to put every element of data in their enterprise into a warehouse without considering true requirements or delivering a successful solution for many months. The proliferation of personal or departmental data marts is another classic approach that might seem like the correct thing to do initially, but over time leaves IT buried in requests for extracts so that the users can do their own analysis. This situation often leads to multiple copies of (unsecured) data, multiple data replication processes, multiple databases and BI tools, and a loss of control of that data and its meaning.

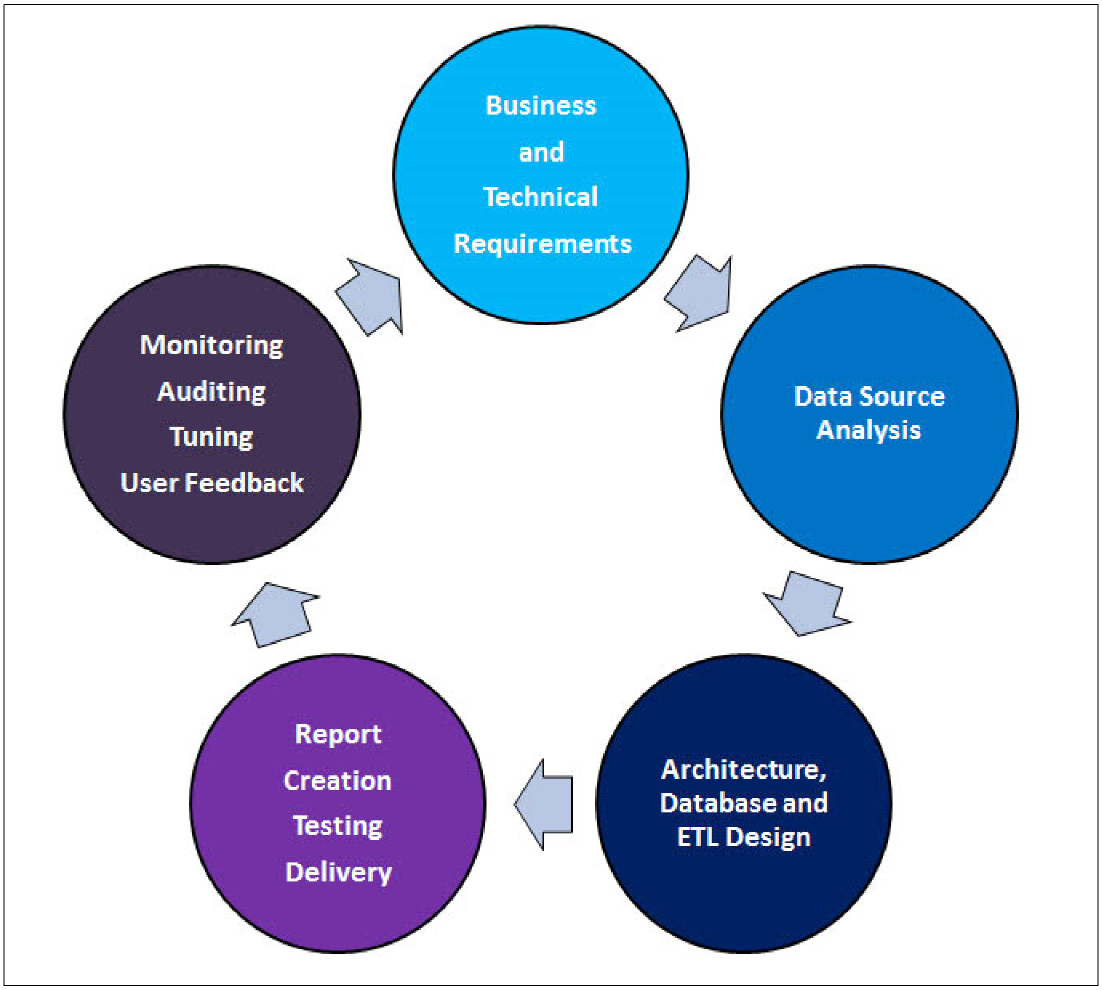

Business and Technical Requirements
Data Source Analysis
Architecture, Database and ETL Design
Report Creation Testing Delivery
Monitoring Auditing Tuning User Feedback

Figure 1-1 The process

1.2 Step 1: Identifying the requirements

It might seem obvious that identifying requirements is the first step. But, it is surprising how many times step one is to pick a visualization tool because of a canned demo with a “wow” factor. Unfortunately, this approach often leaves IT scrambling to provide an infrastructure to support that tool versus building an infrastructure that is the foundation for any tool.

Requirements consist of both business requirements and IT infrastructure requirements. The kinds of questions to ask to draw out requirements might include the following ones:

- How large is your budget?
- What are the questions that the business wants answers to in business terms?
- What are the expectations about user response time?
- What are the expectations about timeliness of the data? Do you need data on demand, instantaneously, or data as recent as last night or last week?
- What are the expectations about receiving reports, for example, by email versus users running reports on demand?
- What are the requirements regarding report consumption devices (for example, mobile device support)?
- What are the requirements for the type of report that is delivered (dashboards, auto drill-down, spreadsheets, or PDFs)?

Current system and personnel requirements are important as well. For example:

- What are average and peak utilizations for CPU, memory, and disks on current systems?
- What is the current HA or DR strategy and processes?
- Is there a plan for growth?
- What data do you have? What data do you not have? Where is that data?
- Who can be the project lead?
- What additional skills must be acquired to support the project?

With requirements in hand, you can start to develop your plan. Getting the business side of the enterprise to agree on the strategy is also paramount to a successful partnership because changing requirements mid-project can lead to a never-ending project.

Using a fictional manufacturer to define a set of high-level requirements as an example, this company has a set of products, suppliers, customers, and salespeople that are employees of the company, and salespeople that act as agents to resell their products. They sell across Canada and the US. They have a limited budget and staff, but know they must get better at simplifying the delivery of information around these types of metrics:

- Daily sales results across all their customers, product sets, and geographies
- Sales versus forecast
- Pricing history
- Sales commission calculations
- Sales trends across various attributes (for example, seasonal trends or weather impacts)
- Promotion planning and tracking
- Salesperson analysis over time

1.3 Data source analysis

Having established the requirements and an understanding of current systems, it is time to dig into the data to understand if and where it exists, and what condition it is in.

The good news for most IBM i shops is that most of the data that is needed for analysis is typically in your DB2 for i databases. However, that is not always the case, and even if it is there, it can sometimes be locked up or hard to access because of legacy database design and standards that were put in place years ago. It is possible that certain rules must be applied to accurately report on a data element. For example, perhaps the fictional manufacturer acquired a company to expand its operations into Canada. At the time, they moved the data into their system, segregated by library. But they did *not* resolve inconsistencies in the data (for example, there are duplicate customer IDs.) So, for example, customer number 05831 in Canada is also a customer number representing a customer in the US, even though they are different customers.

Suppose that sales data from outside sales agents is provided in spreadsheet format, and the employee sales information is in DB2. Perhaps the sales forecast data is stored in a Microsoft SQL Server database because of the opportunity management package that runs there. Other data that can be used to enhance the existing data does not exist, for example, the attachment of a “season of the year” attribute to the data, or acquiring the weather report to supplement existing daily data for future analysis.

What is the timeliness of the data sources? The data in DB2 for i is updated in real time, for example, an order is placed and data is updated. But other data sources are not real time; perhaps agent sales information is received only once a month. Forecast data is not updated in DB2 for i at all, and the only way for the sales manager to compare actual versus forecast data is by using the “gorilla data warehousing” method, which is the manually intensive consolidation of data in a spreadsheet.

In our example, the company lost all the history about which salesperson covered which account because their application and database overwrites the previous salesperson information that was associated with an account when a new representative was assigned.

DB2 Web Query for i is designed with all of these types of common issues in mind.

Web Query is built on a metadata framework, as described in Chapter 4, “Defining metadata” on page 57. The metadata can address anomalies in legacy data that might exist when you deal with that data directly. You can apply rules to the data to extract the key piece of information that is required that is otherwise accessible only by the operational application. In using the metadata approach, you are also creating documentation about your database that might not already exist. Chapter 7, “Accessing additional data sources” on page 141” covers accessing non-DB2 data sources, either in your reports or with DataMigrator.

In addition, DataMigrator, which is covered in Chapter 4, “Defining metadata” on page 57”, contains utilities to profile data to identify possibly dirty data. Further, DataMigrator’s automated extraction of data from various data sources including DB2 for i, Excel spreadsheets, and MS SQL Server (and other heterogeneous relational data-bases, such as MySQL or Oracle) provides a mechanism to consolidate data from many different data sources into a made-for-purpose reporting environment. For example, facilities within DataMigrator allow you to create a surrogate key as a transformation rule to generate a unique key for all customers across Canada and the US. This fictional client could start keeping historical data to know at any point in time which sales representatives covered which customers.

1.4 Designing the solution

Now that you have your requirements set and understand some of the challenges that you have with data, you can move on to designing a solution. Based on the outcome of the data analysis section, it might be time to speak with the project sponsor and project team to indicate where the lack of data might prevent the implementation of something in the original plan. For example, the lack of data pointing to which salesperson covered what accounts in the past is limiting what salesperson analysis you can do. But, by building the historical perspective, you can produce this information in the future. Set expectations early and often.

A key benefit of Web Query is that it can support various architectures that suit your needs based on those requirements that you identified earlier. Which one is correct for you depends on many things, including budget, the need for data consolidation from multiple systems and databases, or the need to isolate the reporting workloads from production systems.

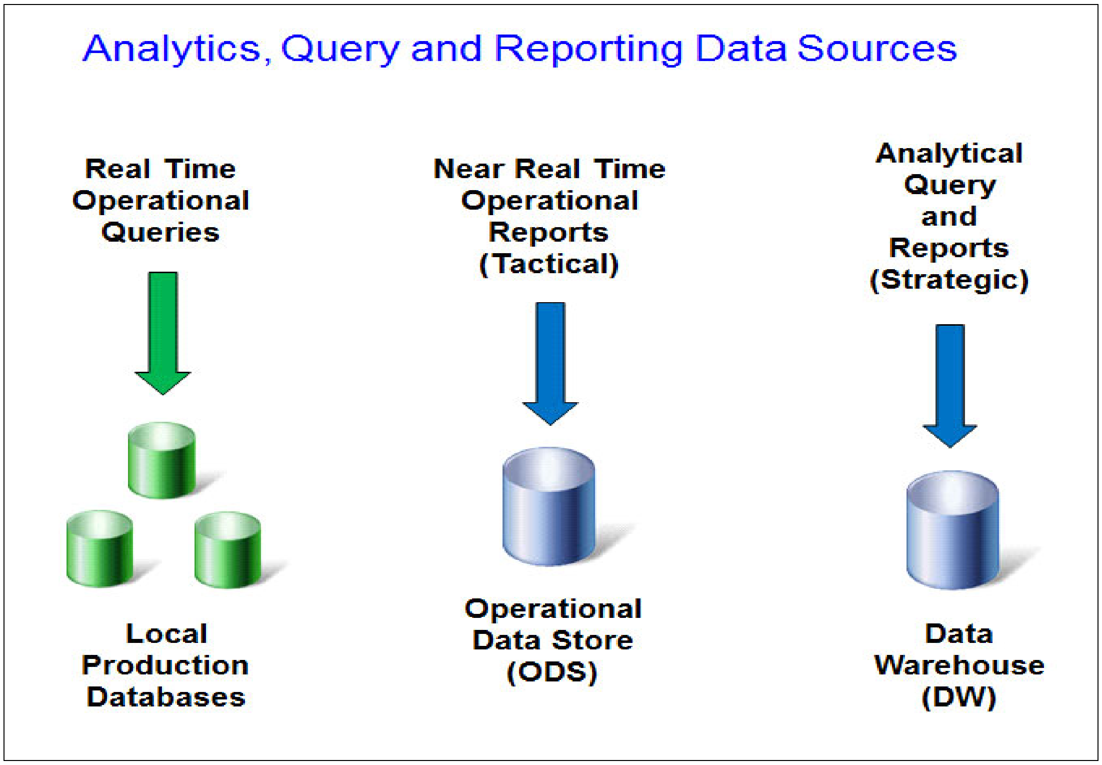
Many people just build reports over their production systems due to the availability of cycles on that system or the need for real-time data access, or because that is where the data is today.

An ODS is typically a near-real time mirrored image of the production databases that is replicated to another server or logical partition. Advantages of this architecture include the isolation of the workload and a consolidation of operational data from different sources. An ODS typically replicates production databases a row at a time, and often in near-real time (log-based replication).

DataMigrator can automate the near-real-time refreshing of an ODS based on its ability to read changed data from database logs (journal receivers) and apply those changes to the target.

The data warehouse architecture can be the ultimate architecture to support analytics because it combines the benefits of the ODS (consolidate data to an isolated LPAR or server) with the ability to transform that data to prepare and optimize for analytics.

Figure 1-2 shows the different architectures that DB2 Web Query for i supports.



Analytics, Query and Reporting Data Sources

Real Time Operational Queries

Local Production Databases

Near Real Time Operational Reports (Tactical)

Operational Data Store (ODS)

Analytical Query and Reports (Strategic)

Data Warehouse (DW)

Figure 1-2 Supporting architectures

DataMigrator provides the Extract, Transform, and Load (ETL) function, which is the cornerstone to automating the building of the data warehouse. Transforming the data is a key element of the data warehouse. It can be used to untangle data that is difficult to access or understand because of legacy database designs, or it can restructure the data from operational databases that are optimized for transaction processing into one that is optimized for analytics.

One example of a common data warehouse data model is the *star-schema* model, which is shown in Figure 1-3.

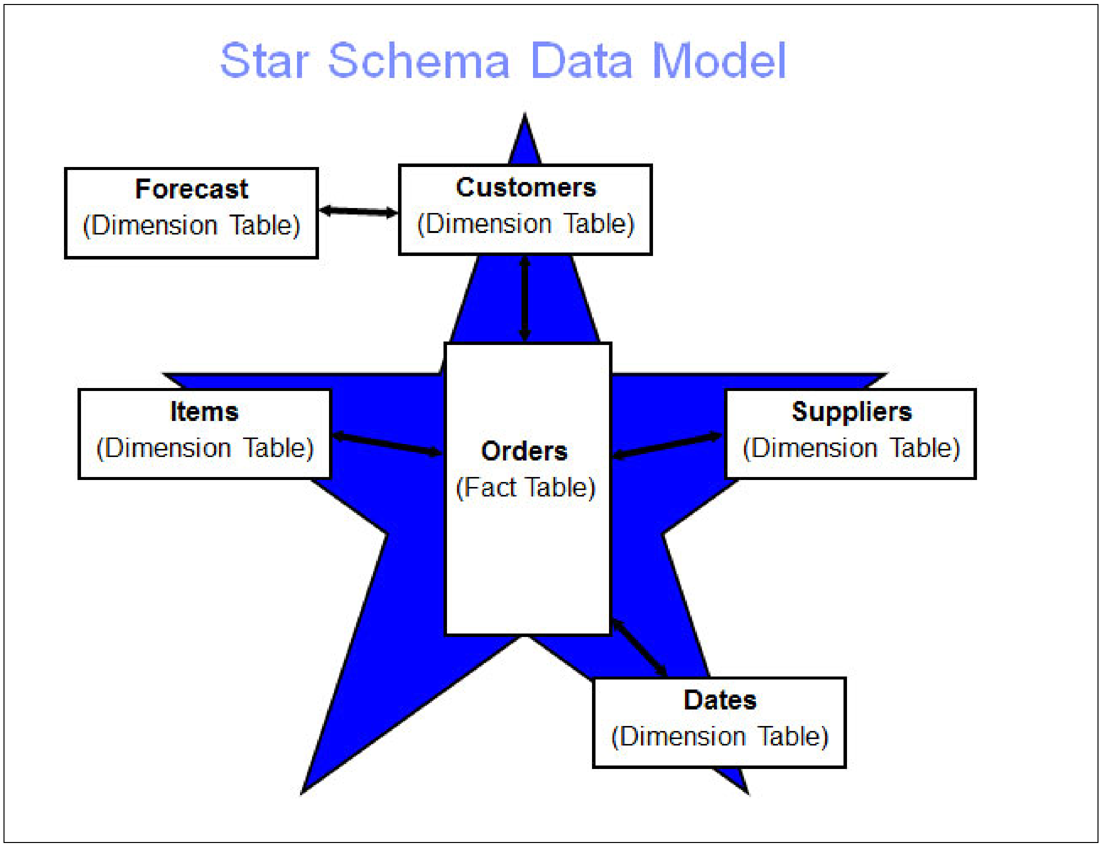


Figure 1-3 Star-schema data model

A properly designed data model can combine with the ETL process to maintain history data, validate data as it enters the warehouse, and automate the building of summary tables for faster performance in the reports.

1.5 Building and deploying reports and dashboards

With the infrastructure defined and clear visibility to the data from the BI tool, you can start building reports or dashboards with Web Query. A number of self guided tutorials are available that contain many learning modules that are focused on how to build reports. These can be found at ibm.co/db2wqwiki.

Most of the clients that the IBM Lab Services team works with have a requirement to deploy information in many ways:

- Spreadsheets, often for business analysts or financial personnel
- Dashboards, for management to quickly assess key performance indicators (KPIs)
- Auto-refreshed dashboards, for sales or trucking operations centers
- Mobile reports for salespeople out in the field
- Drill-down reports for business analysts to view trends or exceptions
- Static PDFs or other output types that can be emailed to certain users

There is not usually a *one size fits all* approach. In addition to the many ways you must deliver information to users, consider the timing of both the data and the report. For many users, a self-service implementation is wanted, meaning that the user can run the report whenever they want. But, must the data be in real time, or is a data snapshot as recent as the end of the previous day okay? What is the user's expectations of response time? Is 1 minute or less acceptable, or less than 5 seconds? Will the user be in a location where they can perform analysis of the data without being connected to your system?

The good news is that Web Query can handle all of these situations by providing many ways that you can build user interfaces and report output types. How can you ensure that you meet user response time requirements? You must construct the correct infrastructure and ensure that your queries are tuned, and that is how DB2 query optimization can help.

1.6 Tuning, monitoring, and auditing

DB2 for i has many strong, self-tuning, and self-managing attributes. Unlike many other database platforms, you do not need several DBAs to manage database performance. But while it is a powerful database engine, it must be optimized to run at its full potential. With the correct SQL optimization work and setting up an administered process that builds reports, you can hopefully meet your user response time requirements.

Although this book does not cover DB2 for i SQL tuning, DB2 for i has many attributes that you can use to improve SQL processes. Because the database access for Web Query reports is through SQL, these attributes can be directly applied.

Here are some of the elements of DB2 for i that Web Query can use:

- Encoded Vector Indexing (EVI), an IBM patented query accelerator:
  - With aggregates
  - EVI Index Only Access
- Maintained temporary indexes
- Database parallelism
- SQL views, user-defined or built-in functions, and SQL procedures
- OLAP and aggregations that are built into DB2 for i (some in OS level 7.3, others that are delivered through previous releases)
- Materialized Query Tables (MQTs)
- Various other aspects that can affect performance, such as using summary tables in your data warehouse, and work management factors, such as dedicated memory allocation for the DB2 Web Query for i subsystem

There are also tips and techniques within Web Query to address performance. One such example is the date dimension table join approach for dealing with date attributes, which is described in Chapter 5, “Date and time functions” on page 69.

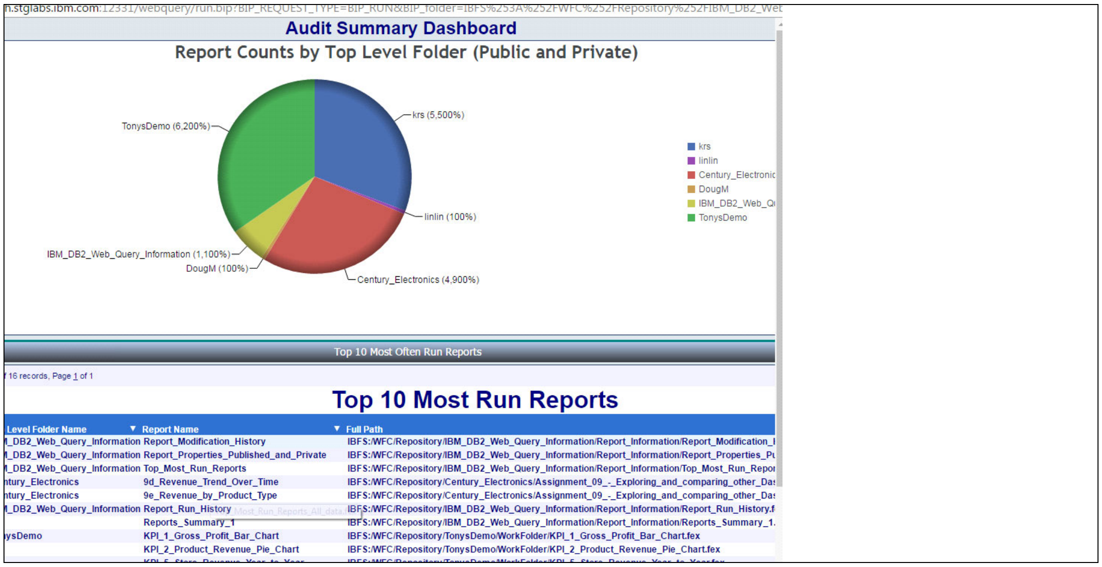
As of Version 2.2 of DB2 Web Query for i, IBM now also delivers a set of reports for monitoring and auditing purposes. These Web Query pre-built reports provide information such as:

- Who is running the most reports?
- Which reports are being run the most?
- Which reports are taking the longest to run?

- How many reports are in each folder?
- When was a report last used?
- When was a report last modified?

These reports are shipped with Web Query as of Version 2.2, hot fix 3 (HF3) fix pack level. For more information about this topic, see the DB2 Web Query for i wiki.

Figure 1-4 shows one of the sample dashboards shipped with Web Query.



n.stglabs.ibm.com:12331/webquery/run.bip?BIP_REQUEST_TYPE=BIP_RUN&BIP_folder=IBFS%253A%252FWFC%252FRepository%252FIBM_DB2_Wet

Audit Summary Dashboard

Report Counts by Top Level Folder (Public and Private)

Top 10 Most Often Run Reports

f 16 records, Page 1 of 1

Top 10 Most Run Reports

Level Folder Name ▼	Report Name ▼	Full Path
I_DB2_Web_Query_Information	Report_Modification_History	IBFS:/WFC/Repository/IBM_DB2_Web_Query_Information/Report_Information/Report_Modification_I
I_DB2_Web_Query_Information	Report_Properties_Published_and_Private	IBFS:/WFC/Repository/IBM_DB2_Web_Query_Information/Report_Information/Report_Properties_Pu
I_DB2_Web_Query_Information	Top_Most_Run_Reports	IBFS:/WFC/Repository/IBM_DB2_Web_Query_Information/Report_Information/Top_Most_Run_Repor
ntury_Electronics	9d_Revenue_Trend_Over_Time	IBFS:/WFC/Repository/Century_Electronics/Assignment_09_-_Exploring_and_comparing_other_Das
ntury_Electronics	9e_Revenue_by_Product_Type	IBFS:/WFC/Repository/Century_Electronics/Assignment_09_-_Exploring_and_comparing_other_Das
I_DB2_Web_Query_Information	Report_Run_History	IBFS:/WFC/Repository/IBM_DB2_Web_Query_Information/Report_Information/Report_Run_History.f
	Reports_Summary_1	IBFS:/WFC/Repository/IBM_DB2_Web_Query_Information/Report_Information/Reports_Summary_1.
ysDemo	KPI_1_Gross_Profit_Bar_Chart	IBFS:/WFC/Repository/TonysDemo/WorkFolder/KPI_1_Gross_Profit_Bar_Chart.fex
	KPI_2_Product_Revenue_Pie_Chart	IBFS:/WFC/Repository/TonysDemo/WorkFolder/KPI_2_Product_Revenue_Pie_Chart.fex
	KPI_5_Store_Revenue_Year_to_Year	IBFS:/WFC/Repository/TonysDemo/WorkFolder/KPI_5_Store_Revenue_Year_to_Year.fex

Figure 1-4 Monitor and Audit reports

1.7 Starting the cycle again

You have successfully completed your first BI project. In many cases, the success of that first project drives the start of the next one, so you will go through the cycle again. However, the second time through the cycle becomes much shorter because of the experience that you gained and processes that you implemented that you can use the second time.

To further assist you in understanding and implementing a BI solution, the IBM DB2 for i Lab Services team offers education and consulting workshops that can provide knowledge transfer, recommendations, and guidance throughout your project. Enablement topics can include the following ones:

- Designing the data warehouse or ODS with proper data modeling techniques
- SQL performance assessment and recommendations
- ETL implementation or guidance with DataMigrator
- DB2 Web Query for i Getting Started Workshop

If you are interested in engaging with the DB2 for i Lab Services team, see DB2 for i Lab Services Offerings.

2

Working with IBM DB2 Web Query for i

This chapter describes DB2 Web Query for i and the benefits of all its various functions. For more information about how to create all the objects that are presented in this chapter, refer to the set of self guided tutorials available from the DB2 Web Query wiki at ibm.co/db2wqwiki.

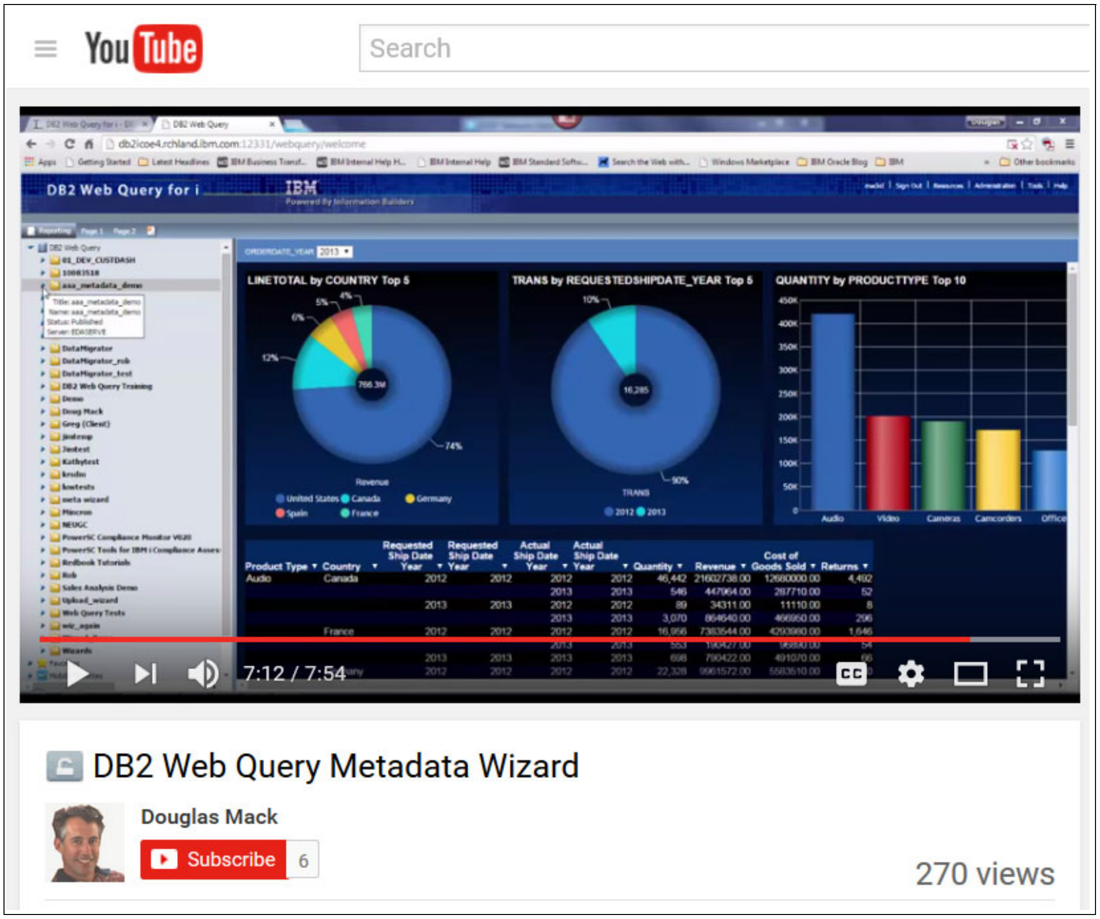
2.1 Videos and important websites

The DB2 Web Query team maintains a couple of websites full of information. The main website, ibm.biz/db2webqueryi contains many useful resources, including Frequently Asked Questions document and links to several short videos. Additional technical information can be found at the DB2 Web Query wiki, ibm.co/db2wqwiki.

At the time of writing, the following videos are available:

- Wizard Analytics Demos
  a. Metadata Wizard
  b. Upload Wizard
- End User Reporting Demonstrations
  a. Online analytical processing (OLAP) Reports
  b. Dashboards
  c. Mobility Support

These videos can also be played from YouTube, as shown in Figure 2-1.



YouTube
Search
DB2 Web Query for i
IBM
LINETOTAL by COUNTRY Top 5
TRANS by REQUESTEDSHIPDATE_YEAR Top 5
QUANTITY by PRODUCTTYPE Top 10
7:12 / 7:54
DB2 Web Query Metadata Wizard
Douglas Mack
Subscribe
6
270 views

Figure 2-1 DB2 Web Query video that is published on YouTube

Note: Here are some additional important websites to refer to as you learn more about DB2 Web Query for i:

- DB2 Web Query Blog
- DB2 for i Blog
- DB2 for i wiki

2.2 Simple reports

The first set of reports are fairly simple and present information that every business might want to track regularly:

1. Revenue and gross summary
2. Top 10 products by margin
3. Revenue summary with subtotals
4. Simple revenue, group by year

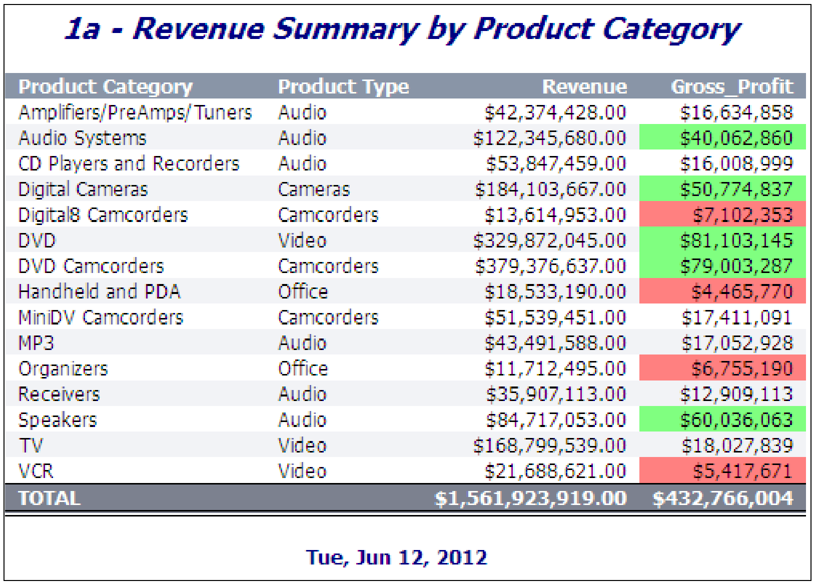
2.2.1 Revenue and gross summary report

The first report is a revenue and gross profit summary. Despite its simplicity, you want to make the report as usable as possible.

The revenue and gross profit report is a fairly basic one that shows the traditional reporting features of grouping and summarization, with a report header and footer to identify the information that is provided and the time that it was extracted from the database (so offline users can be sure of the latest output if multiple runs are compared).

What is unique about this report is that one of the fields, Gross Profit, is a *virtual or defined field*, which is a field that can be defined in the report itself or in metadata. The field’s value is calculated when the report is run based on the information that is stored in the basic data. If the virtual field is defined in metadata, it is available for all reports that are created on the synonym itself, which means that whenever someone uses this Gross Profit field, it is always calculated the same way, thus avoiding the confusion that is generated by different calculation definitions. This is an example of standardization, also known as creating a single version of the truth through metadata.

The Gross Profit field is color-coded based on the summary value for each line by using the *traffic lighting* feature. Traffic lighting helps the report users quickly focus on the good and bad items in the report and identify the top and bottom gross profit performers. The output is the Revenue and Gross Summary Report, as shown in Figure 2-2.



1a - Revenue Summary by Product Category

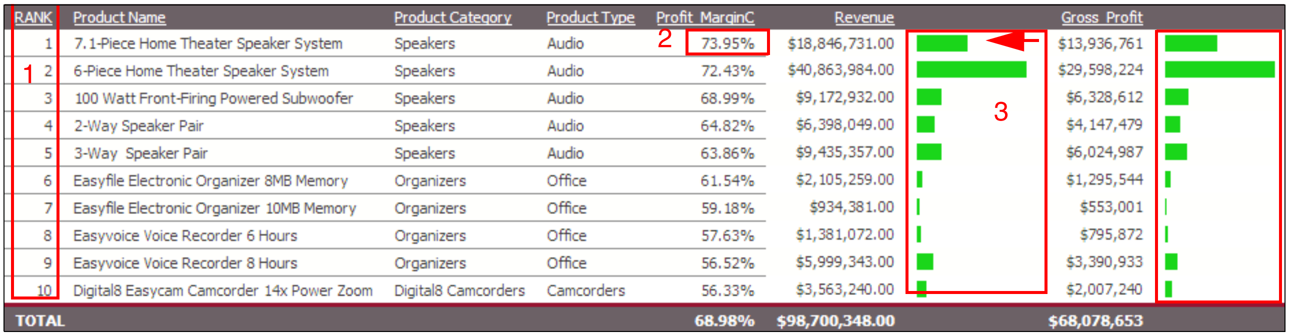
Product Category	Product Type	Revenue	Gross_Profit
Amplifiers/PreAmps/Tuners	Audio	$42,374,428.00	$16,634,858
Audio Systems	Audio	$122,345,680.00	$40,062,860
CD Players and Recorders	Audio	$53,847,459.00	$16,008,999
Digital Cameras	Cameras	$184,103,667.00	$50,774,837
Digital8 Camcorders	Camcorders	$13,614,953.00	$7,102,353
DVD	Video	$329,872,045.00	$81,103,145
DVD Camcorders	Camcorders	$379,376,637.00	$79,003,287
Handheld and PDA	Office	$18,533,190.00	$4,465,770
MiniDV Camcorders	Camcorders	$51,539,451.00	$17,411,091
MP3	Audio	$43,491,588.00	$17,052,928
Organizers	Office	$11,712,495.00	$6,755,190
Receivers	Audio	$35,907,113.00	$12,909,113
Speakers	Audio	$84,717,053.00	$60,036,063
TV	Video	$168,799,539.00	$18,027,839
VCR	Video	$21,688,621.00	$5,417,671
TOTAL		**$1,561,923,919.00**	**$432,766,004**

Tue, Jun 12, 2012

Figure 2-2 Final result of the Revenue and Gross Summary Report

2.2.2 Top 10 products by margin

The next report displays the top 10 product names by profit margin in descending order. You want to identify the most profitable products and see what is their contribution is to the business. This report ranks products by profit margin contributor (1), but also gives insight into the total revenue and gross profit for the top 10 ranking products. Using the *visualization* feature, which is a graphical representation of the measures (revenue and gross profit in this example), it is easy to spot that the product with the highest margin (2) is not the one that generates the highest total revenue (3) in the business. This metric suggests a sales action to increase the top profit margin contributor's revenue, as shown in Figure 2-3.

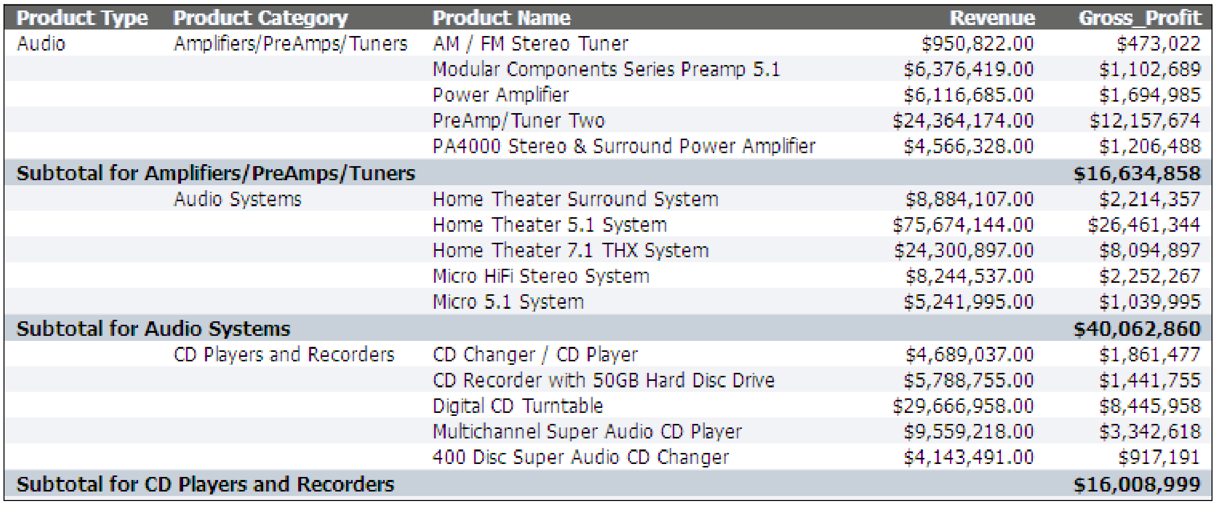


RANK	Product Name	Product Category	Product Type	Profit_MarginC	Revenue	Gross_Profit
1	7.1-Piece Home Theater Speaker System	Speakers	Audio	73.95%	$18,846,731.00	$13,936,761
2	6-Piece Home Theater Speaker System	Speakers	Audio	72.43%	$40,863,984.00	$29,598,224
3	100 Watt Front-Firing Powered Subwoofer	Speakers	Audio	68.99%	$9,172,932.00	$6,328,612
4	2-Way Speaker Pair	Speakers	Audio	64.82%	$6,398,049.00	$4,147,479
5	3-Way Speaker Pair	Speakers	Audio	63.86%	$9,435,357.00	$6,024,987
6	Easyfile Electronic Organizer 8MB Memory	Organizers	Office	61.54%	$2,105,259.00	$1,295,544
7	Easyfile Electronic Organizer 10MB Memory	Organizers	Office	59.18%	$934,381.00	$553,001
8	Easyvoice Voice Recorder 6 Hours	Organizers	Office	57.63%	$1,381,072.00	$795,872
9	Easyvoice Voice Recorder 8 Hours	Organizers	Office	56.52%	$5,999,343.00	$3,390,933
10	Digital8 Easycam Camcorder 14x Power Zoom	Digital8 Camcorders	Camcorders	56.33%	$3,563,240.00	$2,007,240
TOTAL				**68.98%**	**$98,700,348.00**	**$68,078,653**

Figure 2-3 Top 10 product report

2.2.3 Revenue summary with subtotals

One of the most common requests in reports is to break them into predefined levels and obtain summary information (sum, count, average, and minimum) from the levels to gain a perspective about how predefined groups are doing. By using the subtotal feature and setting break points, it is easy to satisfy this request. The third report that is presented in the tutorial displays Gross_Profit subtotals for each product category, as shown in Figure 2-4.

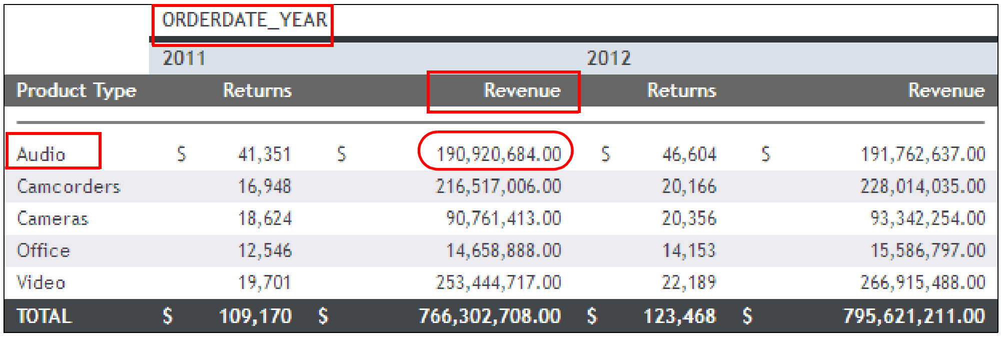


Product Type	Product Category	Product Name	Revenue	Gross_Profit
Audio	Amplifiers/PreAmps/Tuners	AM / FM Stereo Tuner	$950,822.00	$473,022
		Modular Components Series Preamp 5.1	$6,376,419.00	$1,102,689
		Power Amplifier	$6,116,685.00	$1,694,985
		PreAmp/Tuner Two	$24,364,174.00	$12,157,674
		PA4000 Stereo & Surround Power Amplifier	$4,566,328.00	$1,206,488
Subtotal for Amplifiers/PreAmps/Tuners				**$16,634,858**
	Audio Systems	Home Theater Surround System	$8,884,107.00	$2,214,357
		Home Theater 5.1 System	$75,674,144.00	$26,461,344
		Home Theater 7.1 THX System	$24,300,897.00	$8,094,897
		Micro HiFi Stereo System	$8,244,537.00	$2,252,267
		Micro 5.1 System	$5,241,995.00	$1,039,995
Subtotal for Audio Systems				**$40,062,860**
	CD Players and Recorders	CD Changer / CD Player	$4,689,037.00	$1,861,477
		CD Recorder with 50GB Hard Disc Drive	$5,788,755.00	$1,441,755
		Digital CD Turntable	$29,666,958.00	$8,445,958
		Multichannel Super Audio CD Player	$9,559,218.00	$3,342,618
		400 Disc Super Audio CD Changer	$4,143,491.00	$917,191
Subtotal for CD Players and Recorders				**$16,008,999**

Figure 2-4 Revenue Summary with Subtotals

2.2.4 Simple revenue report group by year

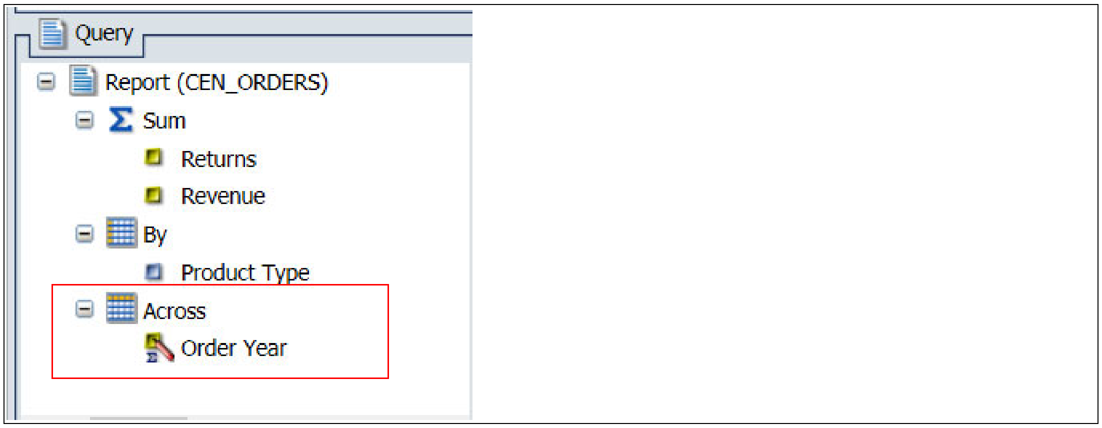
A spreadsheet user is accustomed to getting information that is presented in a matrix style that makes analysis easier. For example, the following request is for summary information about revenue and returns by year and product type by using a pivot report, as shown in Figure 2-5.



	ORDERDATE_YEAR			
	2011		2012	
Product Type	**Returns**	**Revenue**	**Returns**	**Revenue**
Audio	$ 41,351	$ 190,920,684.00	$ 46,604	$ 191,762,637.00
Camcorders	16,948	216,517,006.00	20,166	228,014,035.00
Cameras	18,624	90,761,413.00	20,356	93,342,254.00
Office	12,546	14,658,888.00	14,153	15,586,797.00
Video	19,701	253,444,717.00	22,189	266,915,488.00
TOTAL	**$ 109,170**	**$ 766,302,708.00**	**$ 123,468**	**$ 795,621,211.00**

Figure 2-5 Simple Revenue Report Grouped by Year

The report in Figure 2-5 on page 15 was obtained by using DB2 Web Query by setting the Year dimension to be used *across* the report, as shown in Figure 2-6. The resulting report can be presented to the user directly in Excel or it can be emailed, all in one action.



Query
Report (CEN_ORDERS)
Sum
Returns
Revenue
By
Product Type
Across
Order Year

Figure 2-6 Adding a year dimension

2.3 Date functions in reports

Date information is a key component of most reporting solutions. You want to have information about how you are doing for a year, quarter, and month, or you want to compare how you are doing now as compared to a previous period or to a given target for the period. DB2 Web Query offers various date functions to assist with the presentation of data and complement the powerful date functions that are available in DB2 for i.

Most legacy IBM i applications do not store dates in true ISO date data type fields that DB2 for i supports. That implies that date functions supported in DB2 for i or DB2 Web Query are of little value, unless you can convert that field into a true date field.

You can learn more about dealing with dates in Chapter 5, “Date and time functions” on page 69 but note that DB2 Web Query provides facilities to be able to support powerful date conversion and date attribute processing applicable to many reports - through metadata and use of virtual fields. Here are a few of those approaches:

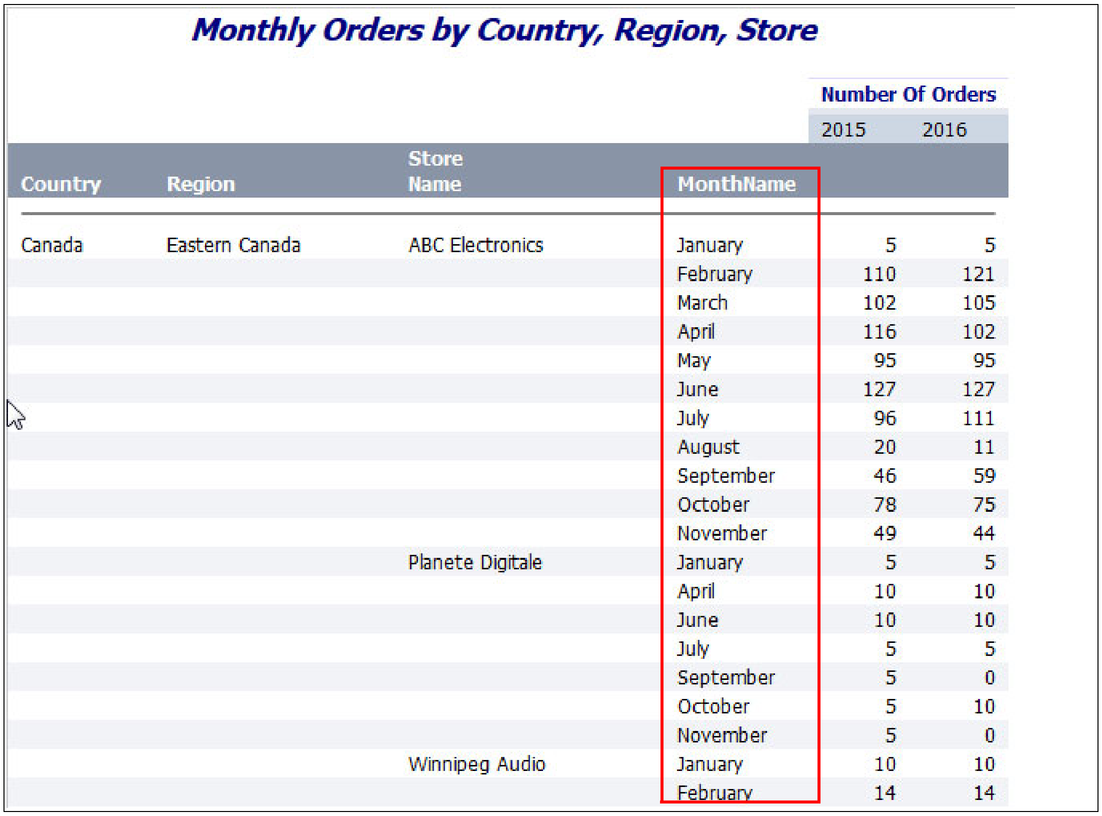
1. Dates using DB2 Web Query functions
2. Dates using an SQL prefix
3. Dates using a date dimension table

2.3.1 Dates using DB2 Web Query for i functions

In this assignment, the report must determine the *number of orders* that are collected in each store for *each month of the year* over the last 2 years. The users want to have the month name exposed in the report, not a number for each month (for example, January instead of 01 and February instead of 02.). Information should be sorted by *country, region*, *store name*, *and name of the month*. The number of orders that are placed must be aggregated and presented by *year*.

Having information that is presented side by side (by year) allows the report consumer to easily compare the numbers year to year.

Data that is stored in the database has no “Order_month_name” column. This information must be obtained applying a specific function to the original date information. An example report is shown in Figure 2-7.



Monthly Orders by Country, Region, Store

				Number Of Orders	
				2015	2016
Country	Region	Store Name	MonthName		
Canada	Eastern Canada	ABC Electronics	January	5	5
			February	110	121
			March	102	105
			April	116	102
			May	95	95
			June	127	127
			July	96	111
			August	20	11
			September	46	59
			October	78	75
			November	49	44
		Planete Digitale	January	5	5
			April	10	10
			June	10	10
			July	5	5
			September	5	0
			October	5	10
			November	5	0
		Winnipeg Audio	January	10	10
			February	14	14

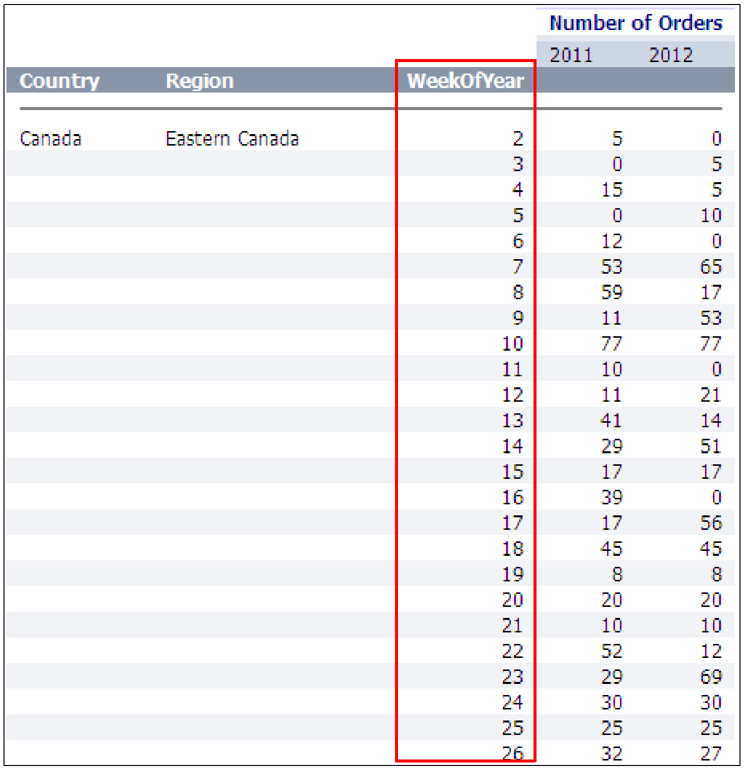
Figure 2-7 Report Result by using the DB2 Web Query for i Date function

2.3.2 Dates using an SQL prefix

The user asks for similar information to be provided by week instead of month, which can be done by using the WEEK function in DB2 SQL.

Note: The WEEK function returns an integer 1 - 54 that represents the week of the year.

DB2 Web Query can specify a direct SQL function and pass that request directly through to the DB2 engine by specifying an SQL prefix in a Define field. In this example, this technique is used to calculate the week of the year for each of the rows in the ORDERS table, as shown in Figure 2-8.



			Number of Orders	
			2011	2012
Country	Region	WeekOfYear		
Canada	Eastern Canada	2	5	0
		3	0	5
		4	15	5
		5	0	10
		6	12	0
		7	53	65
		8	59	17
		9	11	53
		10	77	77
		11	10	0
		12	11	21
		13	41	14
		14	29	51
		15	17	17
		16	39	0
		17	17	56
		18	45	45
		19	8	8
		20	20	20
		21	10	10
		22	52	12
		23	29	69
		24	30	30
		25	25	25
		26	32	27

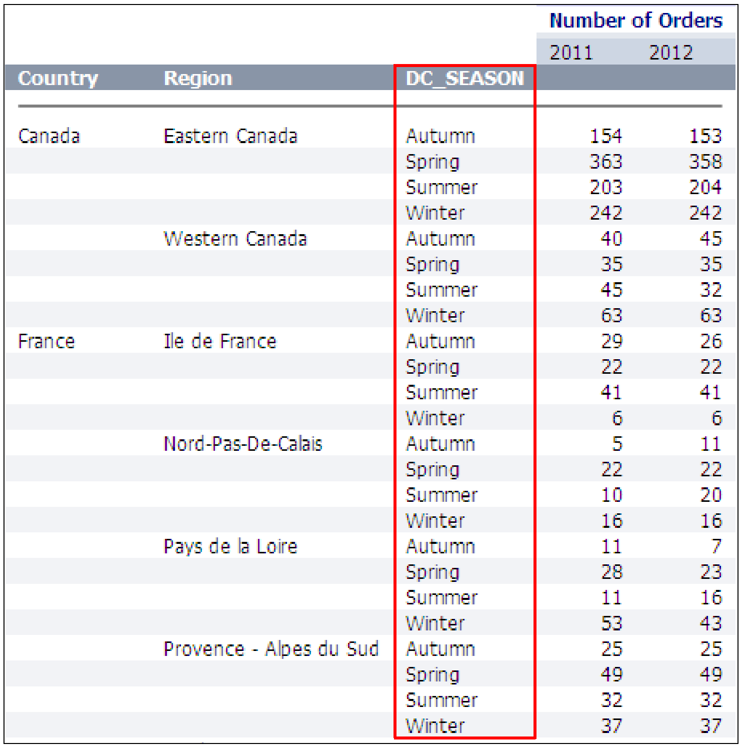
Figure 2-8 Order Date Report Grouped by Week of Year

2.3.3 Dates using a date dimension table

You have seen that it is possible to extract and calculate specific date components by using DB2 Web Query functions and SQL prefixes. However, there are some date requirements that cannot be fulfilled by either DB2 Web Query or SQL functions. For example, what if you wanted to sort and aggregate order dates by the phase of the moon? Or the season of the year? In which season or phase of the moon do you sell the most products? There are no functions in DB2 Web Query for i or SQL that can calculate those components. For specialized requirements like these, a date dimension table is a possible solution.

A date dimension table is a prepopulated table with a row for every day of the year in a broad date range. It contains columns for assorted ways to represent dates. For more information about date dimension tables, see Chapter 5, “Date and time functions” on page 69.

In Figure 2-9, you can see that a season is added in the report by using a date table.



			Number of Orders	
			2011	2012
Country	Region	DC_SEASON		
Canada	Eastern Canada	Autumn	154	153
		Spring	363	358
		Summer	203	204
		Winter	242	242
	Western Canada	Autumn	40	45
		Spring	35	35
		Summer	45	32
		Winter	63	63
France	Ile de France	Autumn	29	26
		Spring	22	22
		Summer	41	41
		Winter	6	6
	Nord-Pas-De-Calais	Autumn	5	11
		Spring	22	22
		Summer	10	20
		Winter	16	16
	Pays de la Loire	Autumn	11	7
		Spring	28	23
		Summer	11	16
		Winter	53	43
	Provence - Alpes du Sud	Autumn	25	25
		Spring	49	49
		Summer	32	32
		Winter	37	37

Figure 2-9 Sample report with Season field

2.4 Various report features and formatting

When using (consuming) reports, the users might face various issues. Imagine receiving a report that is 353 pages long that is about all the items that sold in all shops in your country. But all you wanted to start with is understanding the total revenue by shop and then get to total sales by product for each shop, and then you might want a chart that shows sales by product.

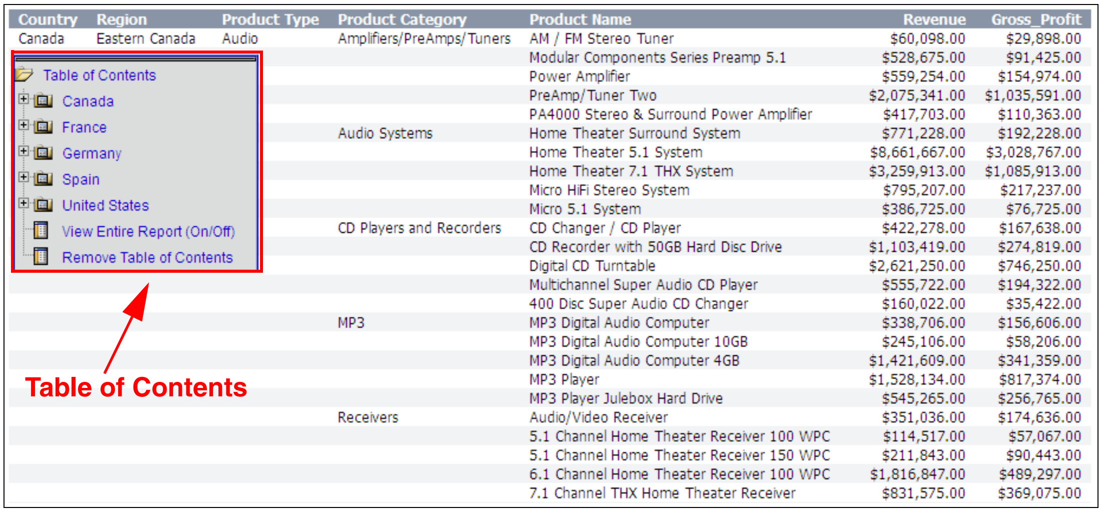
For report usability, it is important to be able to satisfy some basic needs, such as easy navigation in large reports, formatting based on user selection, or analytics capabilities in the report itself. DB2 Web Query offers a rich set of choices:

- Activate the Table of Contents, Accordion, and Pages on Demand features for easy navigation of large reports.
- Generate other report formats, such as PDF, Excel Formula, and Active Reports.
- Provide analytics capabilities through the Active Report and Auto Drill Down formats.

2.4.1 Table of Contents

It is not uncommon to have larger reports that contain many rows of data. Therefore, navigation features are important so that the report consumers can quickly locate information. In our example, data is sorted by Country, Region, Product Type, Product Category, and Product Name. The fields are Revenue and Gross Profit. Because there are so many Sort-By fields, this report is aggregated at many levels and contains many rows in the result set. Providing the report consumer with a quick and easy way to jump to specific locations in the report is the key to acceptance by the users and a successful implementation.

This task can be accomplished by adding a Table of Contents in your report so that you can easily understand the content of your report and navigate from one subset of reports to another one. With the Table of Contents feature activated, report consumers see a window that is initially collapsed to reveal only the first Sort-By field values. Expanding a specific row shows the next level of Sort-By fields. You can use this condensed interface to quickly find specific points of interest in the report and immediately go to that area, as shown in Figure 2-10.

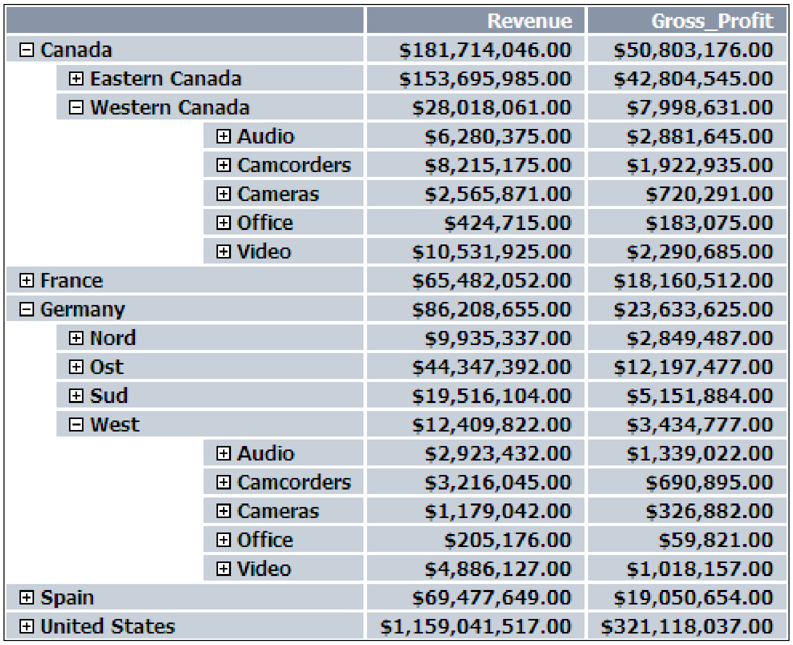


Country	Region	Product Type	Product Category	Product Name	Revenue	Gross_Profit
Canada	Eastern Canada	Audio	Amplifiers/PreAmps/Tuners	AM / FM Stereo Tuner	$60,098.00	$29,898.00
				Modular Components Series Preamp 5.1	$528,675.00	$91,425.00
				Power Amplifier	$559,254.00	$154,974.00
				PreAmp/Tuner Two	$2,075,341.00	$1,035,591.00
				PA4000 Stereo & Surround Power Amplifier	$417,703.00	$110,363.00
			Audio Systems	Home Theater Surround System	$771,228.00	$192,228.00
				Home Theater 5.1 System	$8,661,667.00	$3,028,767.00
				Home Theater 7.1 THX System	$3,259,913.00	$1,085,913.00
				Micro HiFi Stereo System	$795,207.00	$217,237.00
				Micro 5.1 System	$386,725.00	$76,725.00
			CD Players and Recorders	CD Changer / CD Player	$422,278.00	$167,638.00
				CD Recorder with 50GB Hard Disc Drive	$1,103,419.00	$274,819.00
				Digital CD Turntable	$2,621,250.00	$746,250.00
				Multichannel Super Audio CD Player	$555,722.00	$194,322.00
				400 Disc Super Audio CD Changer	$160,022.00	$35,422.00
			MP3	MP3 Digital Audio Computer	$338,706.00	$156,606.00
				MP3 Digital Audio Computer 10GB	$245,106.00	$58,206.00
				MP3 Digital Audio Computer 4GB	$1,421,609.00	$341,359.00
				MP3 Player	$1,528,134.00	$817,374.00
				MP3 Player Julebox Hard Drive	$545,265.00	$256,765.00
			Receivers	Audio/Video Receiver	$351,036.00	$174,636.00
				5.1 Channel Home Theater Receiver 100 WPC	$114,517.00	$57,067.00
				5.1 Channel Home Theater Receiver 150 WPC	$211,843.00	$90,443.00
				6.1 Channel Home Theater Receiver 100 WPC	$1,816,847.00	$489,297.00
				7.1 Channel THX Home Theater Receiver	$831,575.00	$369,075.00

Figure 2-10 Sample report with Table of Contents

2.4.2 Accordion

Another possible solution to get easy navigation in long reports is to initially present aggregated information and allow the user to expand sections as needed. In DB2 Web Query, you can accomplish this task by enabling the *Accordion* format option. In an Accordion report, the user can position, expand, and collapse each Sort-By field level (measure columns are automatically summed at each of the Sort-By levels). When Accordion is activated and the report consumer runs the report, the rows of the report themselves are initially collapsed and aggregated results are presented. The user can expand and collapse each aggregated row, as shown in Figure 2-11. This solution differs from Table of Contents because there is no separate dialog window to control this navigation. Accordion reports are an easy and viable way of consuming long reports, combining both aggregated (even at multiple levels) and detailed views of data in a single object.

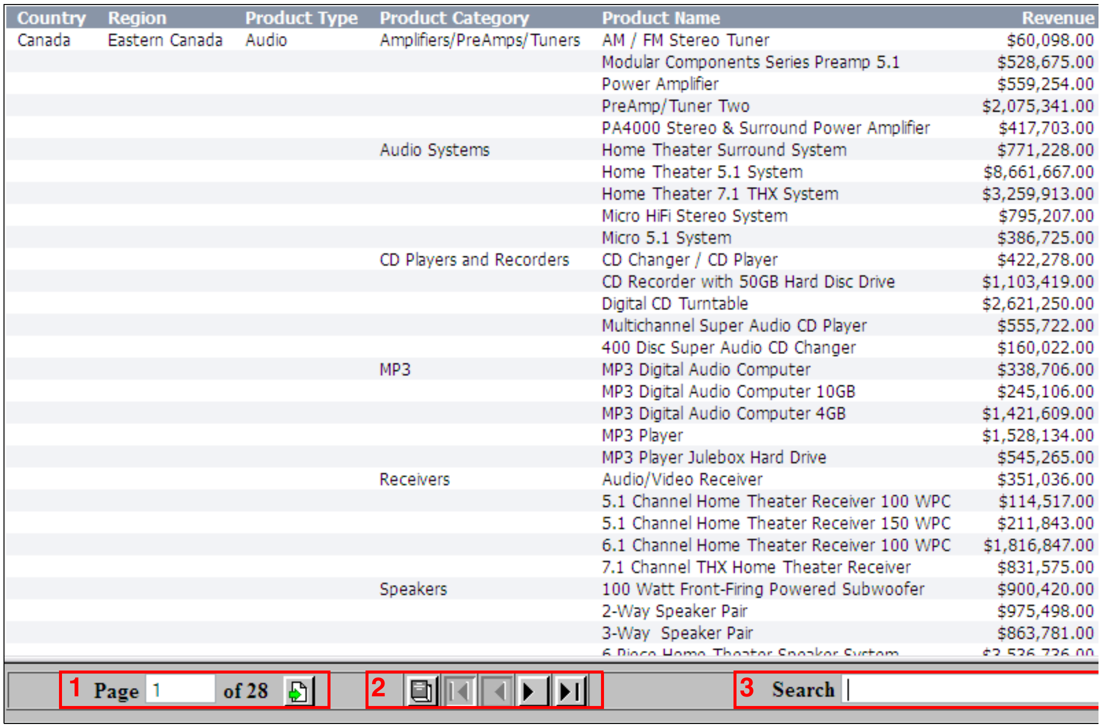


	Revenue	Gross_Profit
⊟ Canada	$181,714,046.00	$50,803,176.00
⊞ Eastern Canada	$153,695,985.00	$42,804,545.00
⊟ Western Canada	$28,018,061.00	$7,998,631.00
⊞ Audio	$6,280,375.00	$2,881,645.00
⊞ Camcorders	$8,215,175.00	$1,922,935.00
⊞ Cameras	$2,565,871.00	$720,291.00
⊞ Office	$424,715.00	$183,075.00
⊞ Video	$10,531,925.00	$2,290,685.00
⊞ France	$65,482,052.00	$18,160,512.00
⊟ Germany	$86,208,655.00	$23,633,625.00
⊞ Nord	$9,935,337.00	$2,849,487.00
⊞ Ost	$44,347,392.00	$12,197,477.00
⊞ Sud	$19,516,104.00	$5,151,884.00
⊟ West	$12,409,822.00	$3,434,777.00
⊞ Audio	$2,923,432.00	$1,339,022.00
⊞ Camcorders	$3,216,045.00	$690,895.00
⊞ Cameras	$1,179,042.00	$326,882.00
⊞ Office	$205,176.00	$59,821.00
⊞ Video	$4,886,127.00	$1,018,157.00
⊞ Spain	$69,477,649.00	$19,050,654.00
⊞ United States	$1,159,041,517.00	$321,118,037.00

Figure 2-11 Revenue Report in Accordion mode

2.4.3 Pages on Demand

A third option for navigating large reports is the Pages on Demand feature. With this option, the result set is broken into pages and you see an interface (see Figure 2-12) that you use to go to a specific page (1) or to position at beginning/end (2) on the report. There is also a search feature (3) that (if a match is found) takes you to the first occurrence of the search word.



Country	Region	Product Type	Product Category	Product Name	Revenue
Canada	Eastern Canada	Audio	Amplifiers/PreAmps/Tuners	AM / FM Stereo Tuner	$60,098.00
				Modular Components Series Preamp 5.1	$528,675.00
				Power Amplifier	$559,254.00
				PreAmp/Tuner Two	$2,075,341.00
				PA4000 Stereo & Surround Power Amplifier	$417,703.00
			Audio Systems	Home Theater Surround System	$771,228.00
				Home Theater 5.1 System	$8,661,667.00
				Home Theater 7.1 THX System	$3,259,913.00
				Micro HiFi Stereo System	$795,207.00
				Micro 5.1 System	$386,725.00
			CD Players and Recorders	CD Changer / CD Player	$422,278.00
				CD Recorder with 50GB Hard Disc Drive	$1,103,419.00
				Digital CD Turntable	$2,621,250.00
				Multichannel Super Audio CD Player	$555,722.00
				400 Disc Super Audio CD Changer	$160,022.00
			MP3	MP3 Digital Audio Computer	$338,706.00
				MP3 Digital Audio Computer 10GB	$245,106.00
				MP3 Digital Audio Computer 4GB	$1,421,609.00
				MP3 Player	$1,528,134.00
				MP3 Player Julebox Hard Drive	$545,265.00
			Receivers	Audio/Video Receiver	$351,036.00
				5.1 Channel Home Theater Receiver 100 WPC	$114,517.00
				5.1 Channel Home Theater Receiver 150 WPC	$211,843.00
				6.1 Channel Home Theater Receiver 100 WPC	$1,816,847.00
				7.1 Channel THX Home Theater Receiver	$831,575.00
			Speakers	100 Watt Front-Firing Powered Subwoofer	$900,420.00
				2-Way Speaker Pair	$975,498.00
				3-Way Speaker Pair	$863,781.00
				6 Piece Home Theater Speaker System	$3,526,726.00

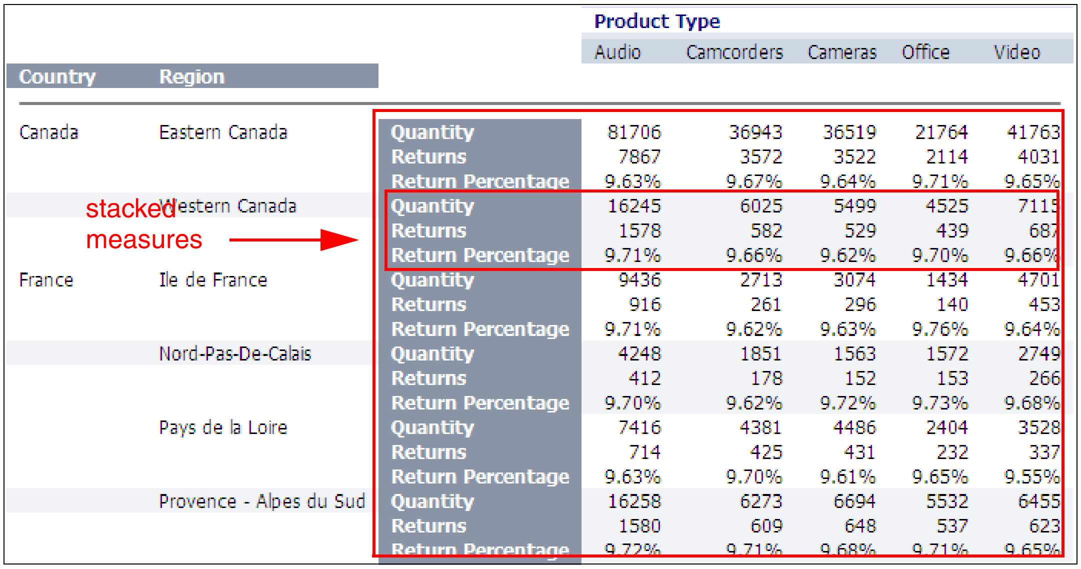
Figure 2-12 Pages on Demand report result

2.4.4 Stack Measures

Sometimes you need to create a report with many measurements (numeric fields), and your reports are too wide to fit onto a screen or part of the columns are printed on a different sheet of paper.

Using the *Stack Measures* feature, it is possible to present measurements (numeric fields) within the same column as lines/rows that are grouped by the various *sort by* elements.

Besides making reports more compact and easy to read and print, you can use this format option to easily compares the values of parameters. You can see the result of enabling the Stack Measure format feature in Figure 2-13.


stacked measures

			Product Type				
Country	Region		Audio	Camcorders	Cameras	Office	Video
Canada	Eastern Canada	Quantity	81706	36943	36519	21764	41763
		Returns	7867	3572	3522	2114	4031
		Return Percentage	9.63%	9.67%	9.64%	9.71%	9.65%
	Western Canada	Quantity	16245	6025	5499	4525	7115
		Returns	1578	582	529	439	687
		Return Percentage	9.71%	9.66%	9.62%	9.70%	9.66%
France	Ile de France	Quantity	9436	2713	3074	1434	4701
		Returns	916	261	296	140	453
		Return Percentage	9.71%	9.62%	9.63%	9.76%	9.64%
	Nord-Pas-De-Calais	Quantity	4248	1851	1563	1572	2749
		Returns	412	178	152	153	266
		Return Percentage	9.70%	9.62%	9.72%	9.73%	9.68%
	Pays de la Loire	Quantity	7416	4381	4486	2404	3528
		Returns	714	425	431	232	337
		Return Percentage	9.63%	9.70%	9.61%	9.65%	9.55%
	Provence - Alpes du Sud	Quantity	16258	6273	6694	5532	6455
		Returns	1580	609	648	537	623
		Return Percentage	9.72%	9.71%	9.68%	9.71%	9.65%

Figure 2-13 Sample report with Stack Measures format feature

2.4.5 Format choices

It is not uncommon that a report might be needed in various formats:

- HTML when it is used in a browser
- PDF when the user wants to save it
- Excel when information must be combined with something coming from a different source
- Active Report for further analysis or customization and enhancements of presentation

You can certainly have multiple versions of the same report with a different format set, but this might make your lists of reports crowded. Over time, you might need to change your reports (for example, when a new data field is needed), and this change can be cumbersome when you must do it to five format versions of the same report. It is much better to have a single instance of the report and allow the user to select the format that is needed when the report is executed.

In DB2 Web Query, the report output format can be predefined during report design, for example, by setting it to HTML, Excel (with Formula or not), PDF, Active Report, OLAP, or the developer can leave it open for user selection at run time. This is done by selecting one of the options that are presented in InfoAssist (the development tool), as shown in Figure 2-14. *User Selection* is one of the choices.

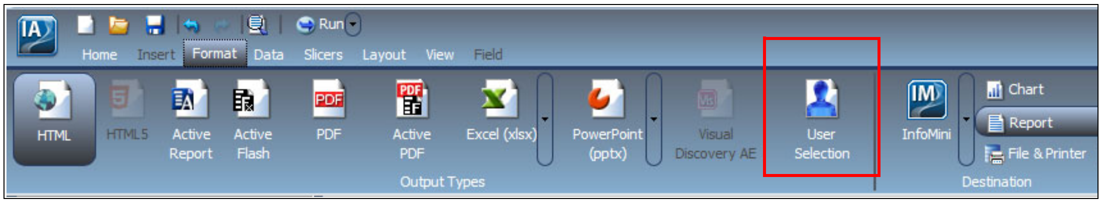


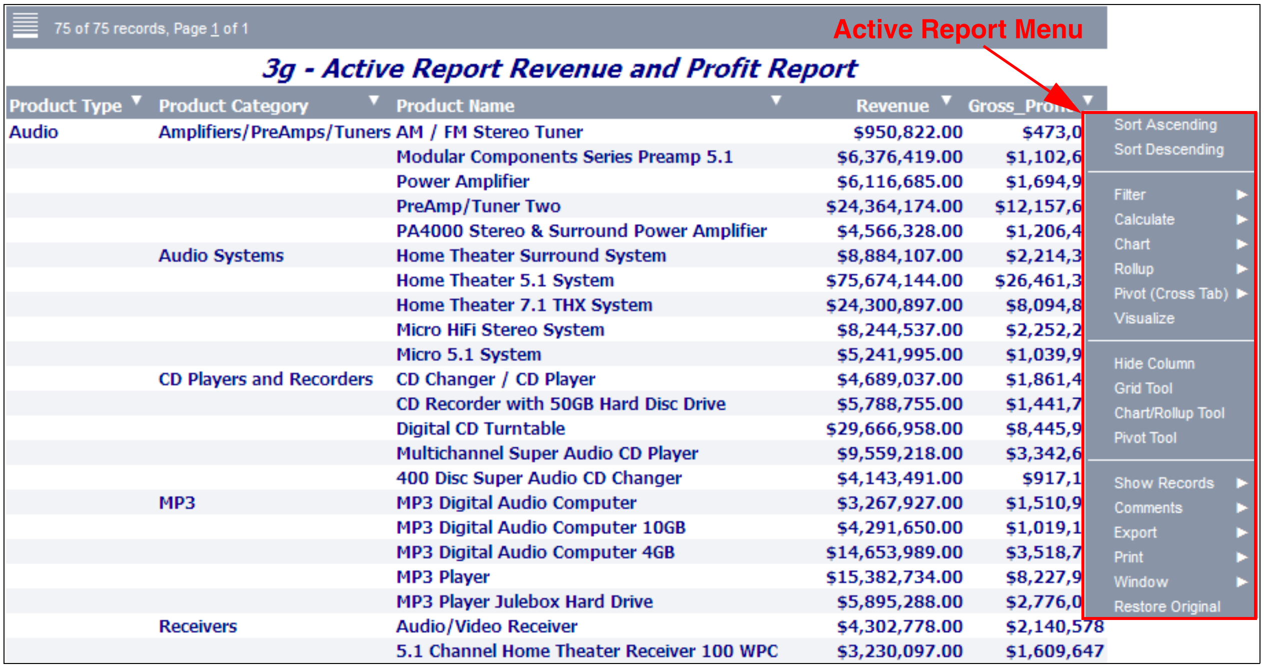
Figure 2-14 InfoAssist Format Options

When User Selection is defined, at run time the user is presented with a set of choices based on the formats that are enabled in the product. The DB2 Web Query administrator can customize this list in the product's console.

Some of the format choices can really enhance the user experience, such as Active Report and Auto Drill Down.

Active Report

When Active Report is selected as the output format, the user gains access to useful analysis capabilities that can allow end users the flexibility of manipulating the views of data in the report. An Active Report with its menu is shown in Figure 2-15.



3g - Active Report Revenue and Profit Report

Product Type	Product Category	Product Name	Revenue	Gross_Profit
Audio	Amplifiers/PreAmps/Tuners	AM / FM Stereo Tuner	$950,822.00	$473,0
		Modular Components Series Preamp 5.1	$6,376,419.00	$1,102,6
		Power Amplifier	$6,116,685.00	$1,694,9
		PreAmp/Tuner Two	$24,364,174.00	$12,157,6
		PA4000 Stereo & Surround Power Amplifier	$4,566,328.00	$1,206,4
	Audio Systems	Home Theater Surround System	$8,884,107.00	$2,214,3
		Home Theater 5.1 System	$75,674,144.00	$26,461,3
		Home Theater 7.1 THX System	$24,300,897.00	$8,094,8
		Micro HiFi Stereo System	$8,244,537.00	$2,252,2
		Micro 5.1 System	$5,241,995.00	$1,039,9
	CD Players and Recorders	CD Changer / CD Player	$4,689,037.00	$1,861,4
		CD Recorder with 50GB Hard Disc Drive	$5,788,755.00	$1,441,7
		Digital CD Turntable	$29,666,958.00	$8,445,9
		Multichannel Super Audio CD Player	$9,559,218.00	$3,342,6
		400 Disc Super Audio CD Changer	$4,143,491.00	$917,1
	MP3	MP3 Digital Audio Computer	$3,267,927.00	$1,510,9
		MP3 Digital Audio Computer 10GB	$4,291,650.00	$1,019,1
		MP3 Digital Audio Computer 4GB	$14,653,989.00	$3,518,7
		MP3 Player	$15,382,734.00	$8,227,9
		MP3 Player Julebox Hard Drive	$5,895,288.00	$2,776,0
	Receivers	Audio/Video Receiver	$4,302,778.00	$2,140,578
		5.1 Channel Home Theater Receiver 100 WPC	$3,230,097.00	$1,609,647

Figure 2-15 Sample of Active Report and its drop-down menu

An Active Report has the following capabilities:

- Sorting
- Filtering
- Calculating (such as sum, average, min, and max)
- Charting
- Rollup
- Pivoting data
- Adding visualization bars
- Exporting

All these capabilities are retained even when the report is used offline, making this format an appealing way to use reports from a mobile device or for quick analysis.

Auto Drill Down Reports

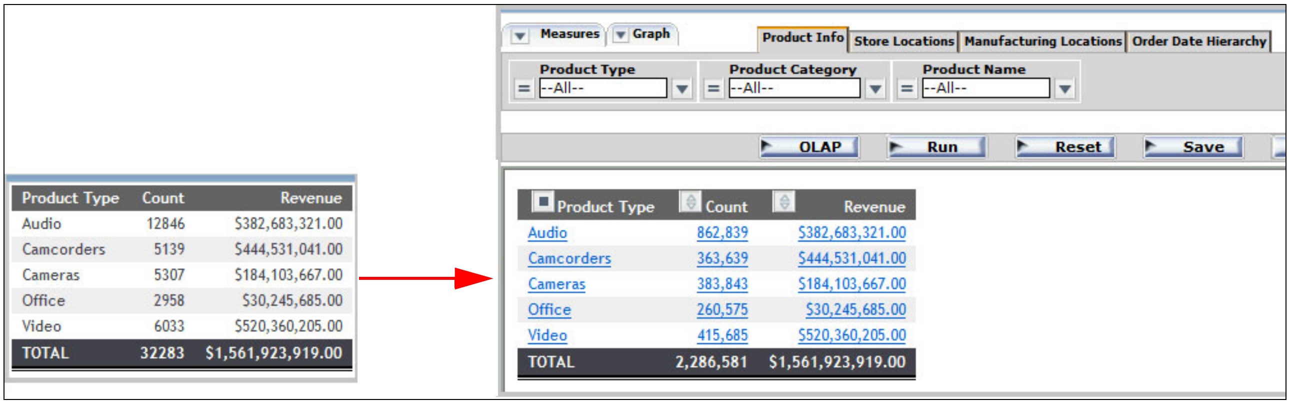
Auto Drill Down reports are the technology that you use to *slice and dice* data or drill down into data. DB2 Web Query makes it easy to provide users with a sophisticated drill-down, slice and dice interface without the report developers having to do anything at all. The developers use a simple enablement flag in the report that automatically provides an interface with drop-down lists and drill-capable columns so that a user can slice and dice or pivot their data and perform various local analysis on the original report.

To use this capability, no action is required to the underlying data - no changes to your tables and views are needed. However, the metadata that presents the data to DB2 Web Query for i must have embedded hierarchies that are predefined either through the Developer Workbench or by using the synonym browser interface. After this task is done, the users have an easy and intuitive way of navigating and drilling down on the hierarchy within their data.

This type of analytics typically starts with a question, such as “What were my regional sales numbers across the United States?” You might look at the numbers and want to see the details for the Eastern region, which are out of line with the other regions. If you then find that a single state is affecting that region’s numbers, then you might ask to see the revenue for that state sorted by the different product groups.

After you find a product group that appears to be underperforming, 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 also referred to as OLAP (Online Analytical Processing). A user can create a separate ad hoc query for each of the above questions, but that is an unlikely action. If a user simply had to click a specific field of interest to go down to the next level of detail, or drag a dimension into the report to get aggregation at that level, or be able to filter on any dimension item with a click, they would be far more likely to continue with their analysis.

A before and after example of an OLAP enabled report is shown in Figure 2-16.

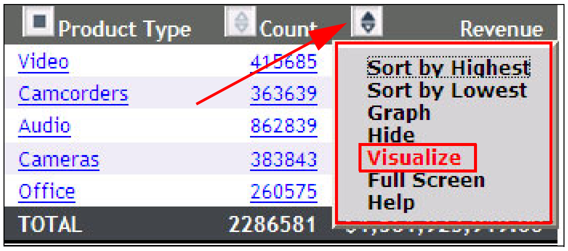


Product Type	Count	Revenue
Audio	12846	$382,683,321.00
Camcorders	5139	$444,531,041.00
Cameras	5307	$184,103,667.00
Office	2958	$30,245,685.00
Video	6033	$520,360,205.00
TOTAL	32283	$1,561,923,919.00

Product Type	Count	Revenue
Audio	862,839	$382,683,321.00
Camcorders	363,639	$444,531,041.00
Cameras	383,843	$184,103,667.00
Office	260,575	$30,245,685.00
Video	415,685	$520,360,205.00
TOTAL	2,286,581	$1,561,923,919.00

Figure 2-16 A simple report (before and after OLAP is enabled)

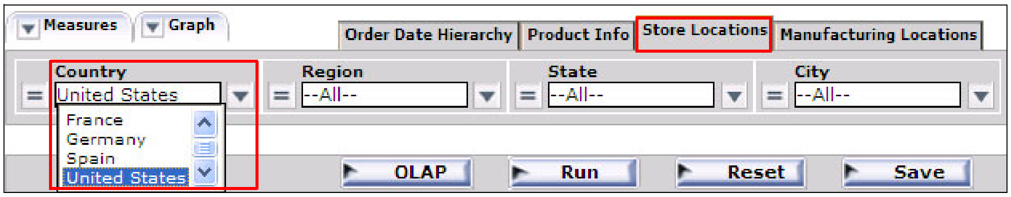
When a report is Auto-Drill-enabled, various actions are allowed. By using the little arrow on the measure columns' title, you open a menu that you can use to act on the column to sort data, create a graph, hide the column, or get a visualization bar on its side. The menu is shown in Figure 2-17.



Product Type	Count	Revenue
Video	415685	
Camcorders	363639	
Audio	862839	
Cameras	383843	
Office	260575	
TOTAL	2286581	

Figure 2-17 Measure column menu

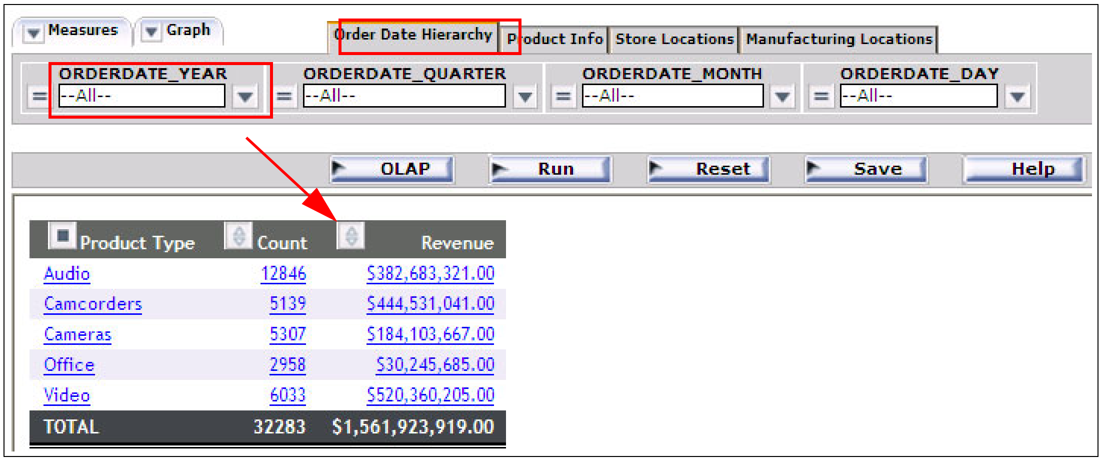
Dimensions can be used to filter and drill down on your data, as shown in Figure 2-18.



Country = United States
Region = --All--
State = --All--
City = --All--

Figure 2-18 Select a hierarchy

You can use the dimensions button to drag more grouping items in your report, as shown in Figure 2-19.

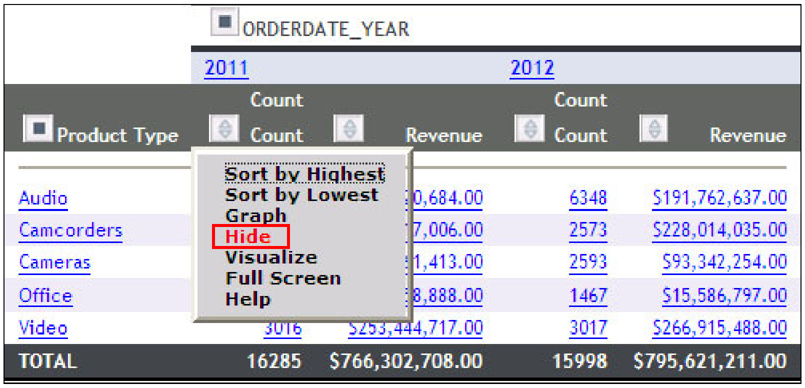


ORDERDATE_YEAR = --All--
ORDERDATE_QUARTER = --All--
ORDERDATE_MONTH = --All--
ORDERDATE_DAY = --All--

Product Type	Count	Revenue
Audio	12846	$382,683,321.00
Camcorders	5139	$444,531,041.00
Cameras	5307	$184,103,667.00
Office	2958	$30,245,685.00
Video	6033	$520,360,205.00
TOTAL	32283	$1,561,923,919.00

Figure 2-19 Drag dimensions

In the resulting report, you can apply more functions, as shown in Figure 2-20, until you get to the needed result.



	ORDERDATE_YEAR			
	2011		2012	
	Count		Count	
Product Type	Count	Revenue	Count	Revenue
Audio		0,684.00	6348	$191,762,637.00
Camcorders		7,006.00	2573	$228,014,035.00
Cameras		1,413.00	2593	$93,342,254.00
Office		8,888.00	1467	$15,586,797.00
Video	3016	$253,444,717.00	3017	$266,915,488.00
TOTAL	16285	$766,302,708.00	15998	$795,621,211.00

Figure 2-20 Hide a field

2.5 Charting

DB2 Web Query provides many different data visualization options. The available graphing, charting, and mapping options range from various bar and line charts to histograms and area charts, stock charts, gauges for key performance indicators, maps, and so on.

This section shows some simple examples:

- Bar chart that shows On Time Delivery (OTD) by Product Type
- Vertical stacked area chart that shows Orders Backlog by Period
- Gauge chart that shows OTD overall
- Map with Revenue by State

2.5.1 Bar chart: on time delivery by product type

A shipping company may want to provide information about on-time or late deliveries. Customers can be dissatisfied when deliveries are too late: They might complain about a late delivery or decide to shop with competitors for their next purchase.

Therefore, one of the main goals of this example is to measure when orders are delivered as scheduled (OTD) versus when they are delivered late. In the process, tracking for early delivery should be accomplished.

Figure 2-21 shows a bar chart that shows OTD measures. OTD analysis is broken down into the following three categories:

- Early: Actual ship date was 10 days or more before the requested ship date.
- On time: Actual ship date was within 10 days of (before or after) the requested ship date.
- Late: Actual ship date was 10 days or more after the requested ship date.

For each Product Type, the percentage breakdown for each of these three categories is shown.

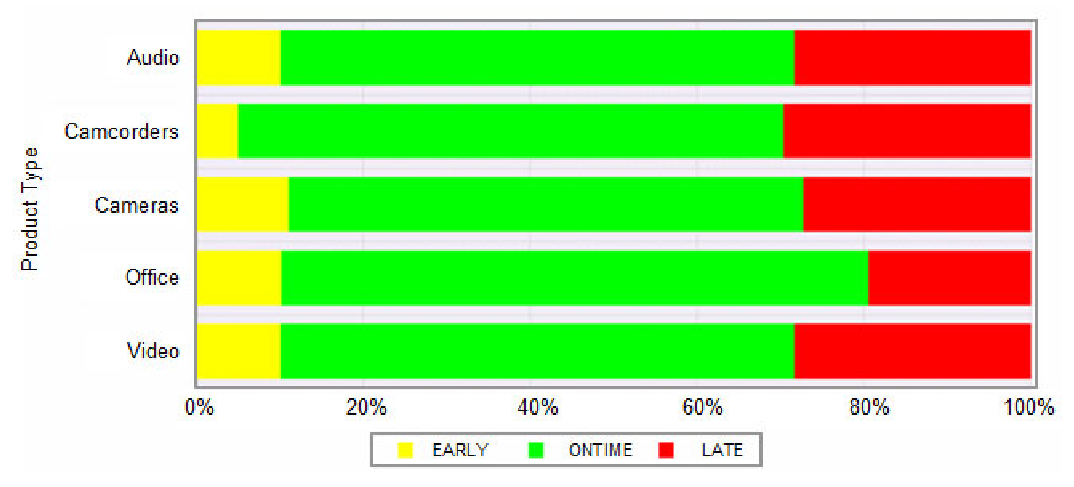


Figure 2-21 Bar chart: On Time Delivery by Product Type

2.5.2 Vertical stacked area: orders backlog by period

The order backlog must be checked each month. A vertical stacked area chart that shows orders backlog by period for each product type is idea for this task, as shown in Figure 2-22. It is more effective than simply presenting rows and columns of numbers.

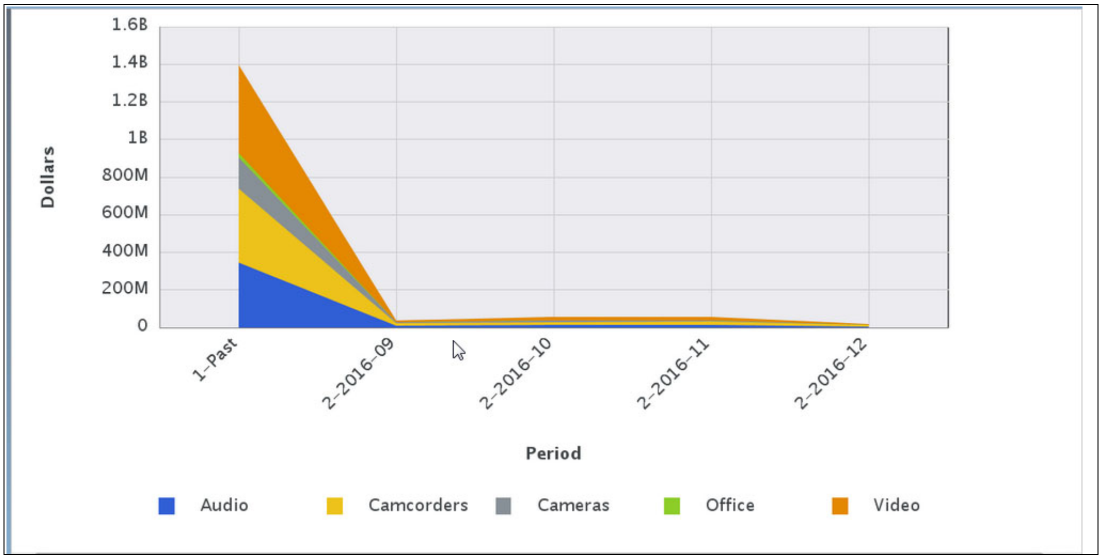


Figure 2-22 Vertical Stacked Area chart: Orders Backlog by Period

2.5.3 Gauge chart: on time delivery overall

Gauge charts are ideal when it is necessary to show what is the actual performance compared to a predefined target.

You want to get OTD information that shows the overall company-wide OTD percentage. The following categories are used to measure and analyze performance in this area:

- Poor: Less than 60%
- Average: 60% - 79%
- Good: Greater than or equal to 80%

Our goal in this area is 85% on-time delivery.

A gauge chart is a popular and efficient way to present a single measure along with various bands to represent poor, average, and good performance. A red band is typically used to show the poor range, yellow to represent the average range, and green to display the good range. You can even identify a goal and show that as a blue band in the gauge as well.

The OTD overall gauge chart is shown in Figure 2-23.

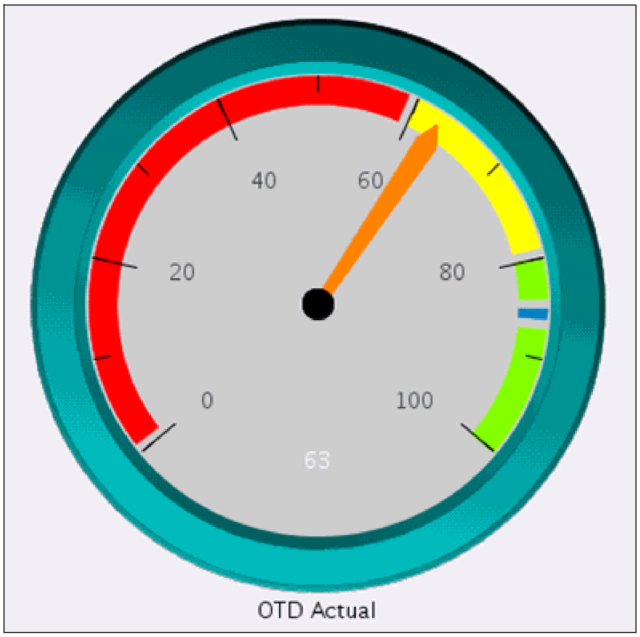


Figure 2-23 Gauge chart: On Time Delivery Overall

2.5.4 Map: revenue by state

Most businesses have some geographical connotation. DB2 Web Query comes with many geographical maps that can give an immediate perception of where your business takes place from a location standpoint. For example, a map chart identifies US revenue distribution by state. This particular bubble chart shows that the east coast states are great contributors to the business, and on the west coast, California dominates.

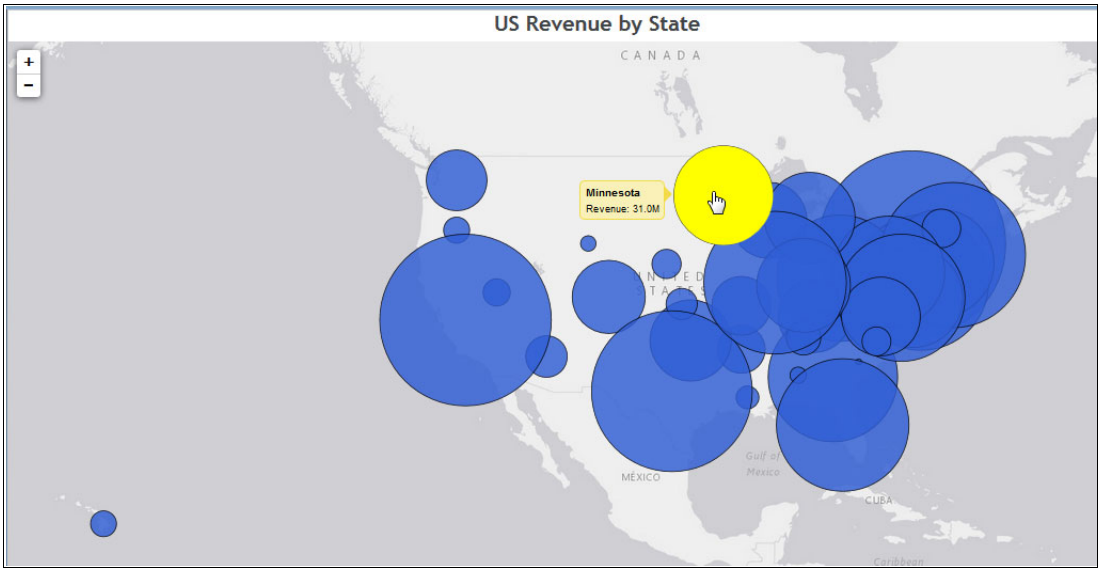


Figure 2-24 Map chart: Revenue by state

2.6 Filters

On many occasions, you want only specific sets of data, which can be obtained by using **WHERE** clauses in your reports. You can tailor reports to extract specific sets of information at run time and use **WHERE** clauses to restrict the resulting data to what information in which the user is interested. For example, a generic sales or revenue report by product type and category can query different countries and time frames every time it is run and the user may set these parameters when they start the report. Again, the same set of reports can serve various area managers that can customize their selections when submitting the request.

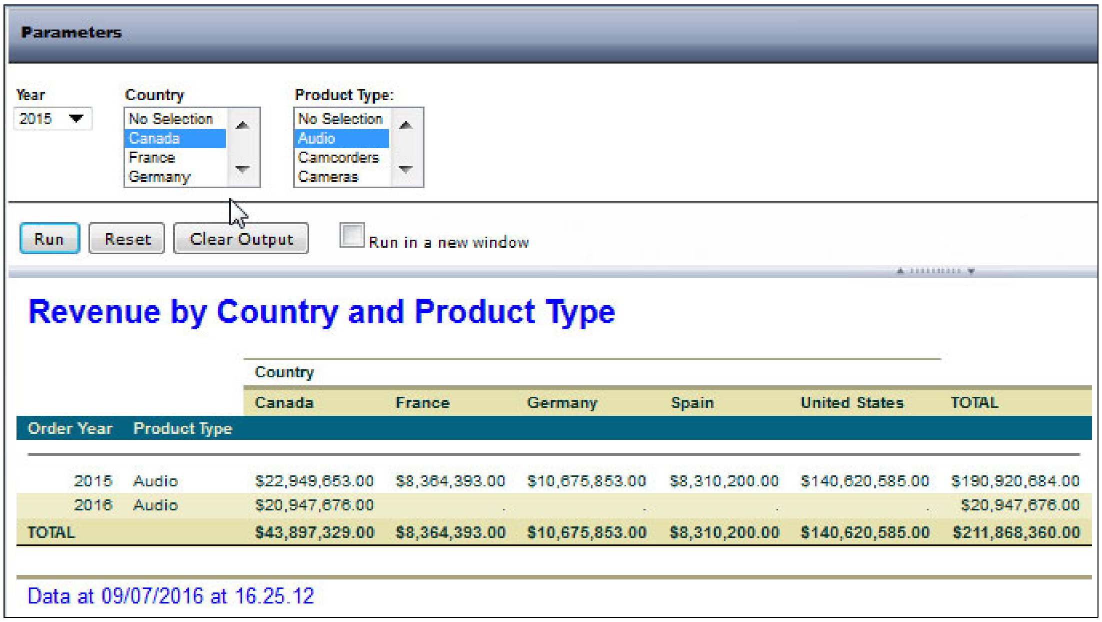
In DB2 Web Query for i, you can accomplish this task by using filters in basic reports and charts, enabling slicers in InfoMini reports, or enabling in reports filters that are defined in metadata.

You can use all these techniques to include the data that you want and exclude the data that you do not want to appear in the report. Filtering options can be predefined and always applied or they can be dynamically set at run time by the user by selecting from lists of possible values that are populated dynamically at run time. Here are two examples:

- Multiple filters in reports and graphs
- InfoMini slicers

2.6.1 Multiple filters in reports and charts

The simple report that is shown in Figure 2-25 summarizes revenue by product type and product category. Each user can select a year, countries, and product types for which they want data by using simple filters with multiple choices. Filters are prepopulated with lists of valid values, which are extracted by the database at run time. The corresponding SQL **`WHERE`** clause is effectively “**`WHERE (T2."PRODUCTTYPE" = 'Audio') AND ((YEAR(T1."ORDERDATE") = 2015) OR (T3."COUNTRY" = 'Canada'))`**”.



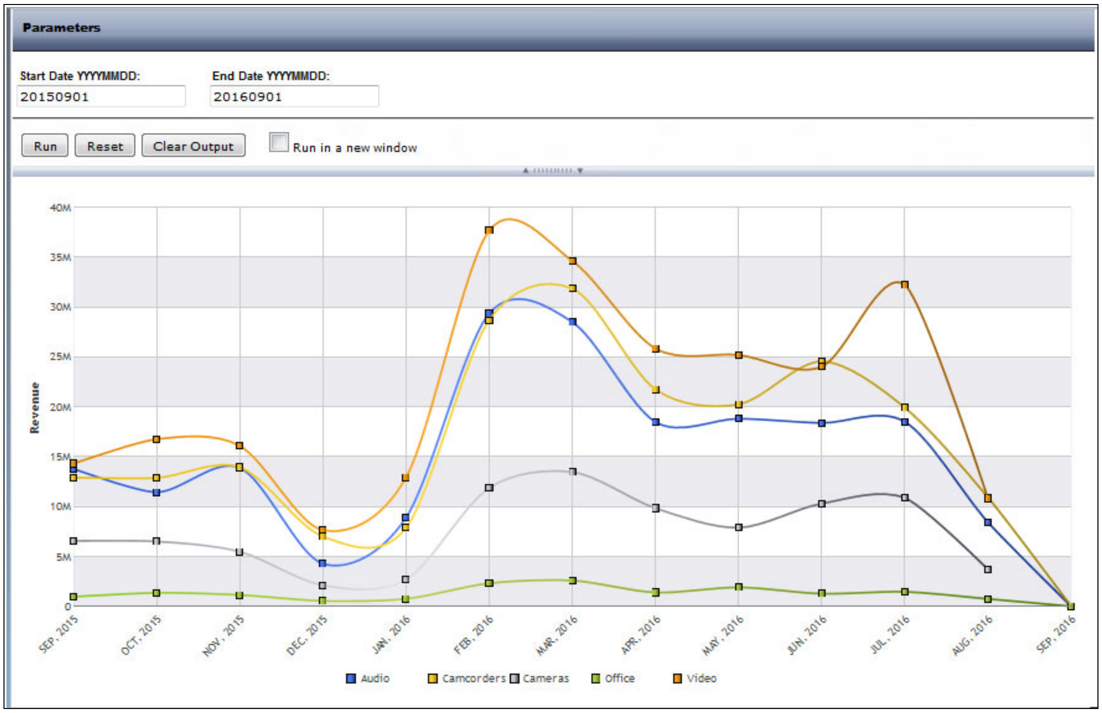
Revenue by Country and Product Type

		Country					
		Canada	France	Germany	Spain	United States	TOTAL
Order Year	Product Type						
2015	Audio	$22,949,653.00	$8,364,393.00	$10,675,853.00	$8,310,200.00	$140,620,585.00	$190,920,684.00
2016	Audio	$20,947,676.00	.	.	.	.	$20,947,676.00
TOTAL		$43,897,329.00	$8,364,393.00	$10,675,853.00	$8,310,200.00	$140,620,585.00	$211,868,360.00

Data at 09/07/2016 at 16.25.12

Figure 2-25 Advanced filters report

Filters can also be applied to charts. Figure 2-26 shows a simple line chart that spans 12 months. The user that runs the report specifies the time range by entering dates in to the provided parameter input fields.



Parameters

Start Date YYYYMMDD: 20150901

End Date YYYYMMDD: 20160901

□ Run in a new window

Figure 2-26 Trending chart with a variable date range

2.6.2 InfoMini slicers

There are instances when flexibility is needed. Some times all the information is needed, and other times only some subset is required, so having predefined filters options might not be the best choice. Greater flexibility is needed to accommodate these requests:

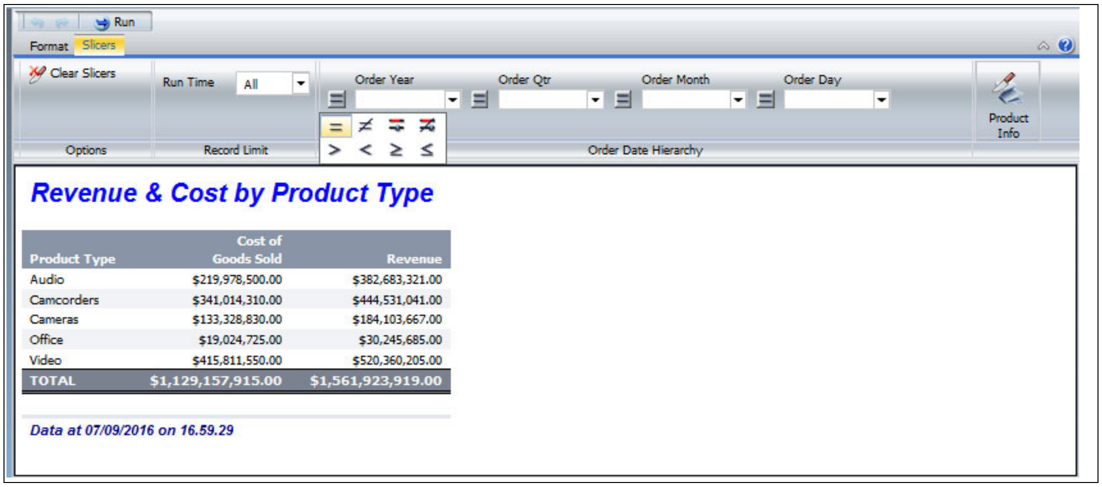
- Setting the comparison operator at run time (such as equal, not equal, and greater than)
- Deciding which filter to populate
- Deciding how to combine multiple filters

In DB2 Web Query, this sort of requirement may be accommodated by using the *InfoMini* report format, which allows flexible filtering options through the Slicers tab and flexible formatting options through its Format tab.

Suppose that you needed a report that accomplishes the following tasks:

- It must summarize revenue and costs by product type.
- It must have the capability to be filtered on all or any of the time elements and on a geographical basis.
- The users must have the capability to set up the compare operators at run time (that is, decide whether they want =, <, <, >=, or others). However, the users must be prevented from choosing the “no data” combination (that is, country=’France’ and city=’Honolulu’ → no Honolulu city in France → 0 records returned from query).

Figure 2-27 shows an example of the requested report, which is obtained by using an InfoMini report with slicers.



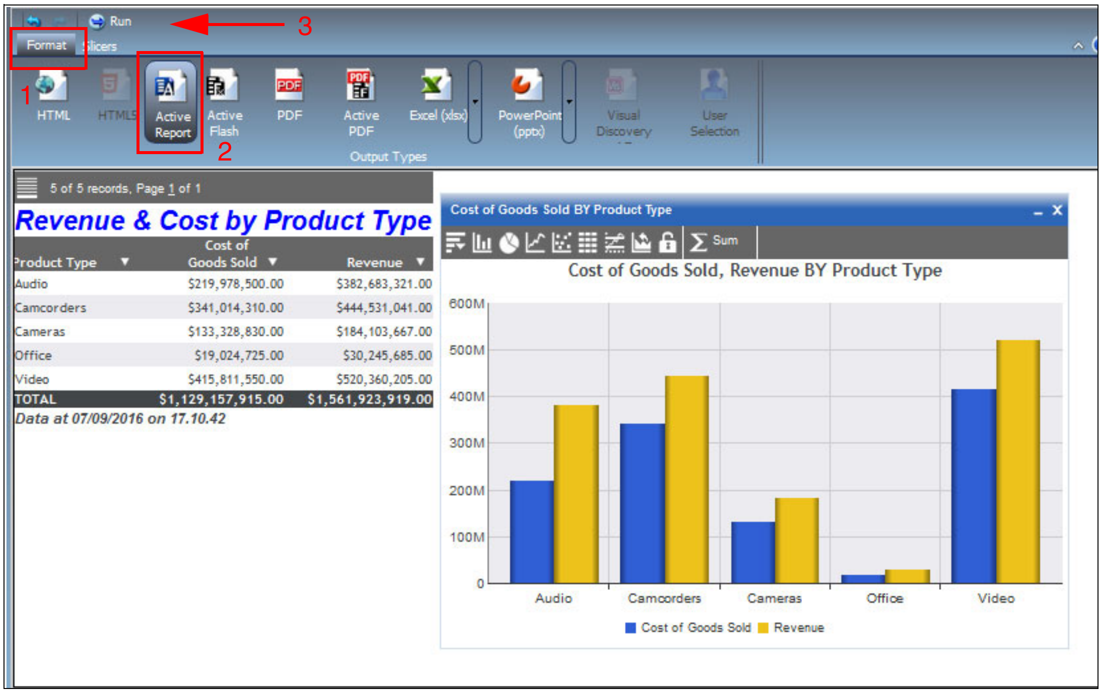
Revenue & Cost by Product Type

Product Type	Cost of Goods Sold	Revenue
Audio	$219,978,500.00	$382,683,321.00
Camcorders	$341,014,310.00	$444,531,041.00
Cameras	$133,328,830.00	$184,103,667.00
Office	$19,024,725.00	$30,245,685.00
Video	$415,811,550.00	$520,360,205.00
TOTAL	$1,129,157,915.00	$1,561,923,919.00

Data at 07/09/2016 on 16.59.29

Figure 2-27 InfoMini report with slicers tab

InfoMini also has a Format tab that the user can use to select the format at run time, as shown in Figure 2-28 at (1). In this example, the Active Flash format was selected (2). This format provides many features, including the capability to chart data. The user then clicks **Run** (3) to run the report.



Revenue & Cost by Product Type

Product Type ▼	Cost of Goods Sold ▼	Revenue ▼
Audio	$219,978,500.00	$382,683,321.00
Camcorders	$341,014,310.00	$444,531,041.00
Cameras	$133,328,830.00	$184,103,667.00
Office	$19,024,725.00	$30,245,685.00
Video	$415,811,550.00	$520,360,205.00
TOTAL	$1,129,157,915.00	$1,561,923,919.00

Data at 07/09/2016 on 17.10.42

Figure 2-28 InfoMini report with the format tab

2.7 Drilling down

In many cases, a user does not want to be overwhelmed with information because it is difficult to focus on the key areas. They might prefer aggregate information that they can navigate to in order to find details about the meaningful items when the need arises, or even better, receive different and detailed information according to the situation that emerges from aggregated information.

To satisfy that need, a user can use a report to report hyperlink drill down technique, with conditional actions that depend on what is found in the aggregated information.

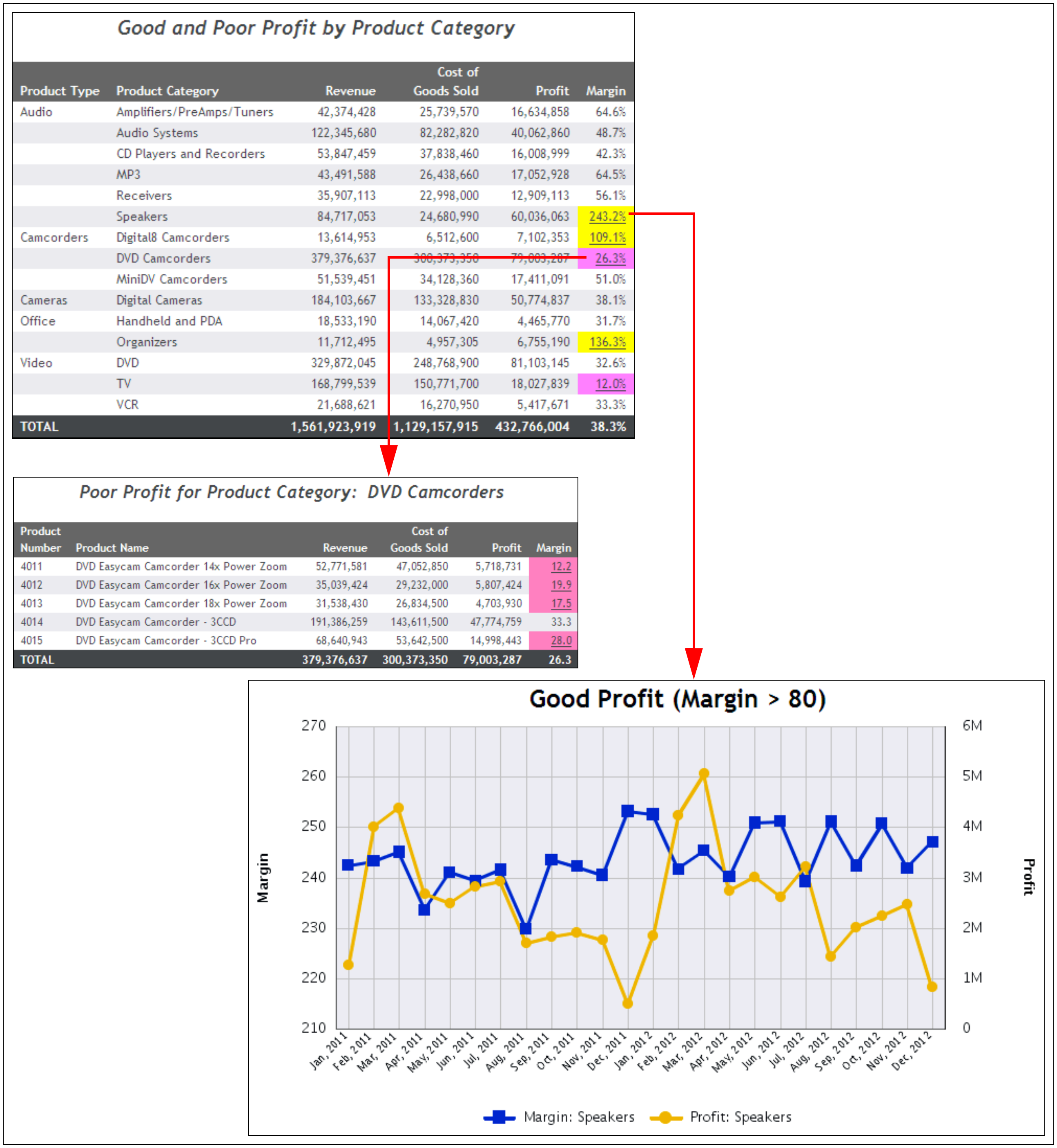
You can use DB2 Web Query to drill down from a report (or chart) to another report (or chart), and passing parameters if the child report requires them. Hyperlinks can be defined to perform different actions according to the condition that is met in the linked field. This scenario has the following parameters:

- Good and Poor Profit by Product Category main report.
- A Good Profit chart is called when a good profit cell is selected.
- A Poor Profit report is called when a poor profit cell is selected.

In the main report, if you click a “Good Profit” link, a chart is created that shows the margin and the profit of the selected (good profit) product category.

When you click a “Poor Profit” link, the second report is created that shows all the products of the selected Product Category.

This flow of information is shown in Figure 2-29.



Good and Poor Profit by Product Category

Product Type	Product Category	Revenue	Cost of Goods Sold	Profit	Margin
Audio	Amplifiers/PreAmps/Tuners	42,374,428	25,739,570	16,634,858	64.6%
	Audio Systems	122,345,680	82,282,820	40,062,860	48.7%
	CD Players and Recorders	53,847,459	37,838,460	16,008,999	42.3%
	MP3	43,491,588	26,438,660	17,052,928	64.5%
	Receivers	35,907,113	22,998,000	12,909,113	56.1%
	Speakers	84,717,053	24,680,990	60,036,063	243.2%
Camcorders	Digital8 Camcorders	13,614,953	6,512,600	7,102,353	109.1%
	DVD Camcorders	379,376,637	300,373,350	79,003,287	26.3%
	MiniDV Camcorders	51,539,451	34,128,360	17,411,091	51.0%
Cameras	Digital Cameras	184,103,667	133,328,830	50,774,837	38.1%
Office	Handheld and PDA	18,533,190	14,067,420	4,465,770	31.7%
	Organizers	11,712,495	4,957,305	6,755,190	136.3%
Video	DVD	329,872,045	248,768,900	81,103,145	32.6%
	TV	168,799,539	150,771,700	18,027,839	12.0%
	VCR	21,688,621	16,270,950	5,417,671	33.3%
TOTAL		1,561,923,919	1,129,157,915	432,766,004	38.3%

Poor Profit for Product Category: DVD Camcorders

Product Number	Product Name	Revenue	Cost of Goods Sold	Profit	Margin
4011	DVD Easycam Camcorder 14x Power Zoom	52,771,581	47,052,850	5,718,731	12.2
4012	DVD Easycam Camcorder 16x Power Zoom	35,039,424	29,232,000	5,807,424	19.9
4013	DVD Easycam Camcorder 18x Power Zoom	31,538,430	26,834,500	4,703,930	17.5
4014	DVD Easycam Camcorder - 3CCD	191,386,259	143,611,500	47,774,759	33.3
4015	DVD Easycam Camcorder - 3CCD Pro	68,640,943	53,642,500	14,998,443	28.0
TOTAL		379,376,637	300,373,350	79,003,287	26.3

Figure 2-29   Basic report and drilled-down report and chart

This scenario can be further articulated by adding multiple drill-down levels as needed. Drill-down is a powerful technique that users can use to navigate from summary to detailed information as needed, without the hassle of explicitly running various queries. Detailed information is gathered and automatically input into the parameters prompts of the detail report by a simple mouse selection.

2.8 Documents and dashboards

You might need to use sets of information for monitoring daily business activity, viewing key metrics, spotting trends, or as a trigger to initiate actions for business improvement. These tasks can be accomplished by using higher level dashboards that contain groups of reports or charts. Dashboards can serve as a means of providing centralized reporting and delivering of Key Performance Indicator (KPI) information to executives and upper management!

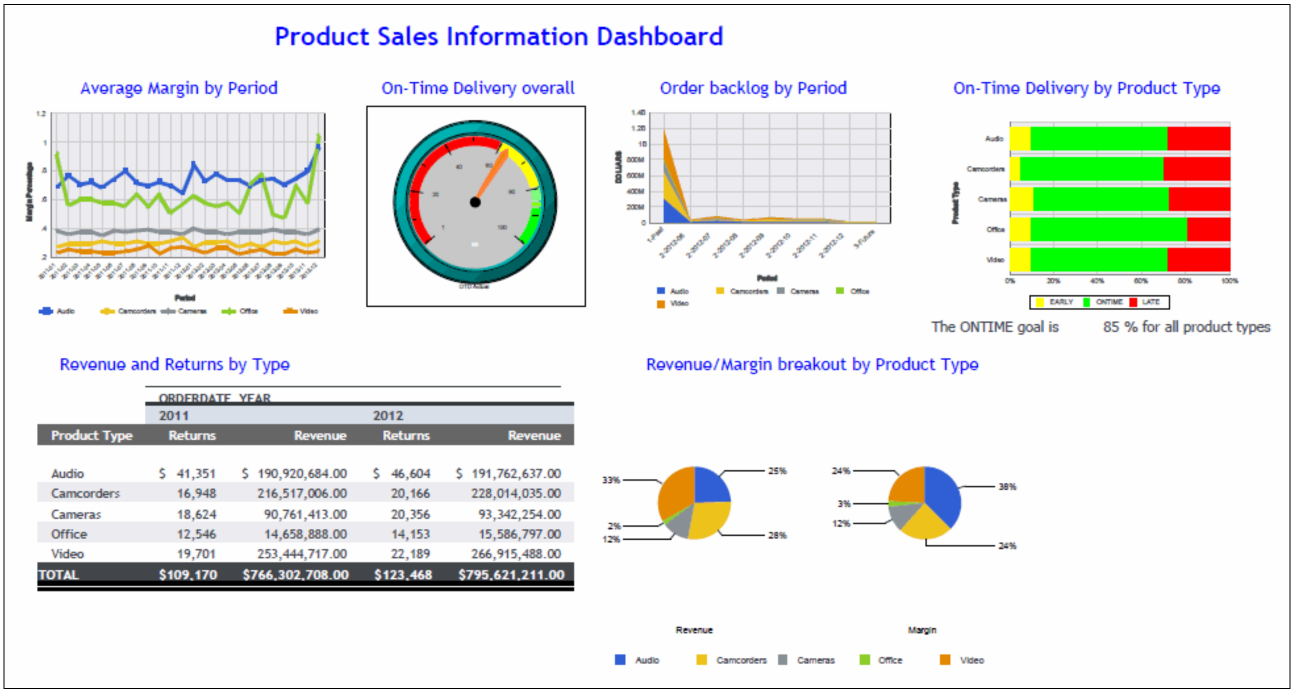
DB2 Web Query for i has many different ways of delivering dashboards:

- Documents and dashboards in InfoAssist
- Developer Workbench HTML Composer
- DB2 Web Query for i Spreadsheet Client

2.8.1 Documents and dashboards in InfoAssist

In InfoAssist, it is easy to build “coordinated documents” and dashboards that present multiple reports and graphs together.

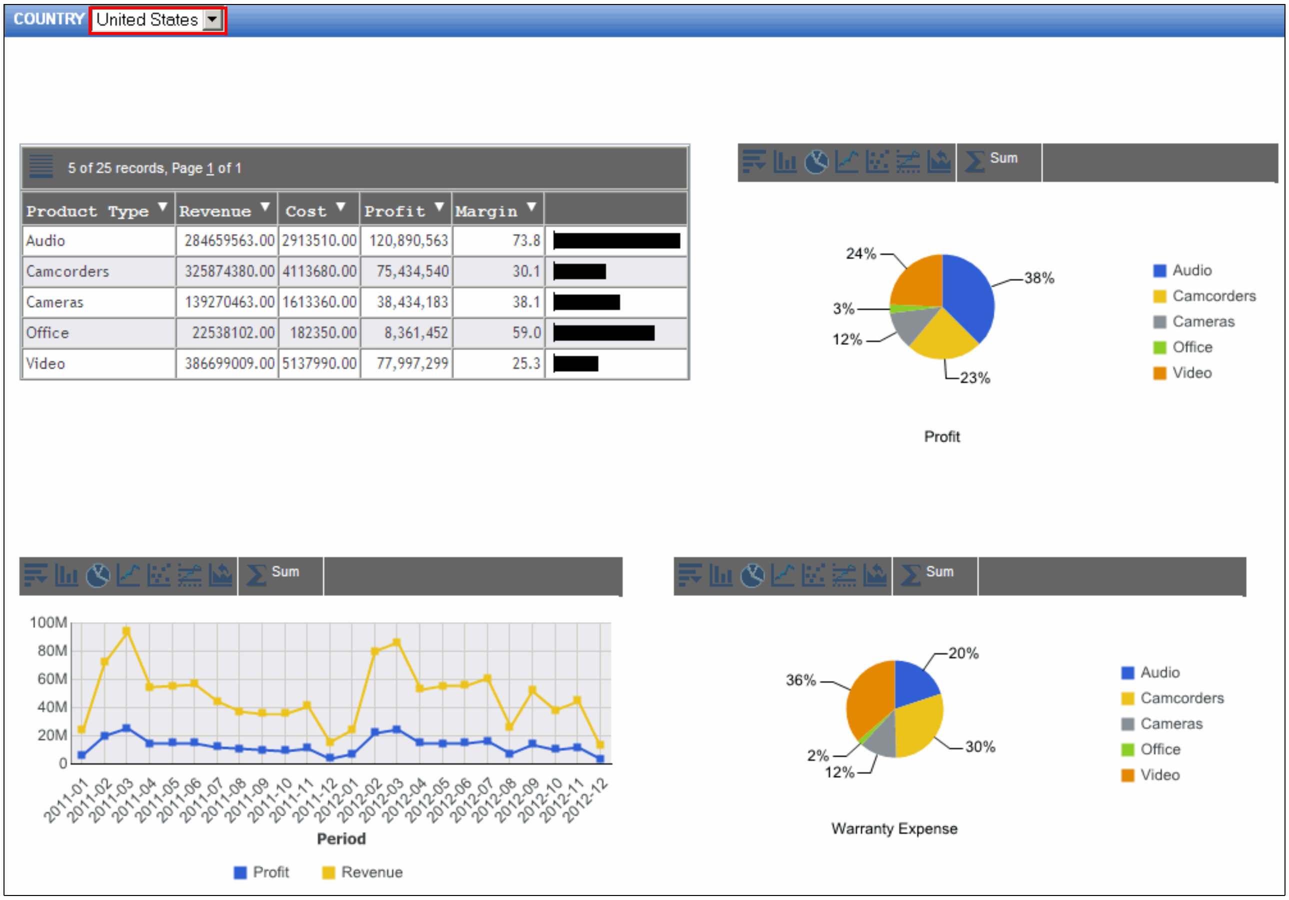
Figure 2-30 shows an example of a dashboard containing several measures of sales performance for product types. This dashboard is developed as a document in DB2 Web Query and shows various charts and reports giving several different views of Product Type information.



	ORDERDATE_YEAR			
	2011		2012	
Product Type	Returns	Revenue	Returns	Revenue
Audio	$ 41,351	$ 190,920,684.00	$ 46,604	$ 191,762,637.00
Camcorders	16,948	216,517,006.00	20,166	228,014,035.00
Cameras	18,624	90,761,413.00	20,356	93,342,254.00
Office	12,546	14,658,888.00	14,153	15,586,797.00
Video	19,701	253,444,717.00	22,189	266,915,488.00
TOTAL	$109,170	$766,302,708.00	$123,468	$795,621,211.00

Figure 2-30 Example document containing charts and reports

A dashboard can also be defined to have all embedded objects show the same subset of data, that is, it can be *coordinated*. Suppose you must create a dashboard to show sales information by country. The country field is the coordinating field and each chart or report in the document specifies country under the **coordinated** section of the Query area. At run time, the coordinated field appears at the top of the window, as shown in Figure 2-31. As the user changes the Country value, the reports and charts change simultaneously, showing the applicable country information in each one.



COUNTRY United States

5 of 25 records, Page 1 of 1

Product Type	Revenue	Cost	Profit	Margin	
Audio	284659563.00	2913510.00	120,890,563	73.8	▬
Camcorders	325874380.00	4113680.00	75,434,540	30.1	▬
Cameras	139270463.00	1613360.00	38,434,183	38.1	▬
Office	22538102.00	182350.00	8,361,452	59.0	▬
Video	386699009.00	5137990.00	77,997,299	25.3	▬

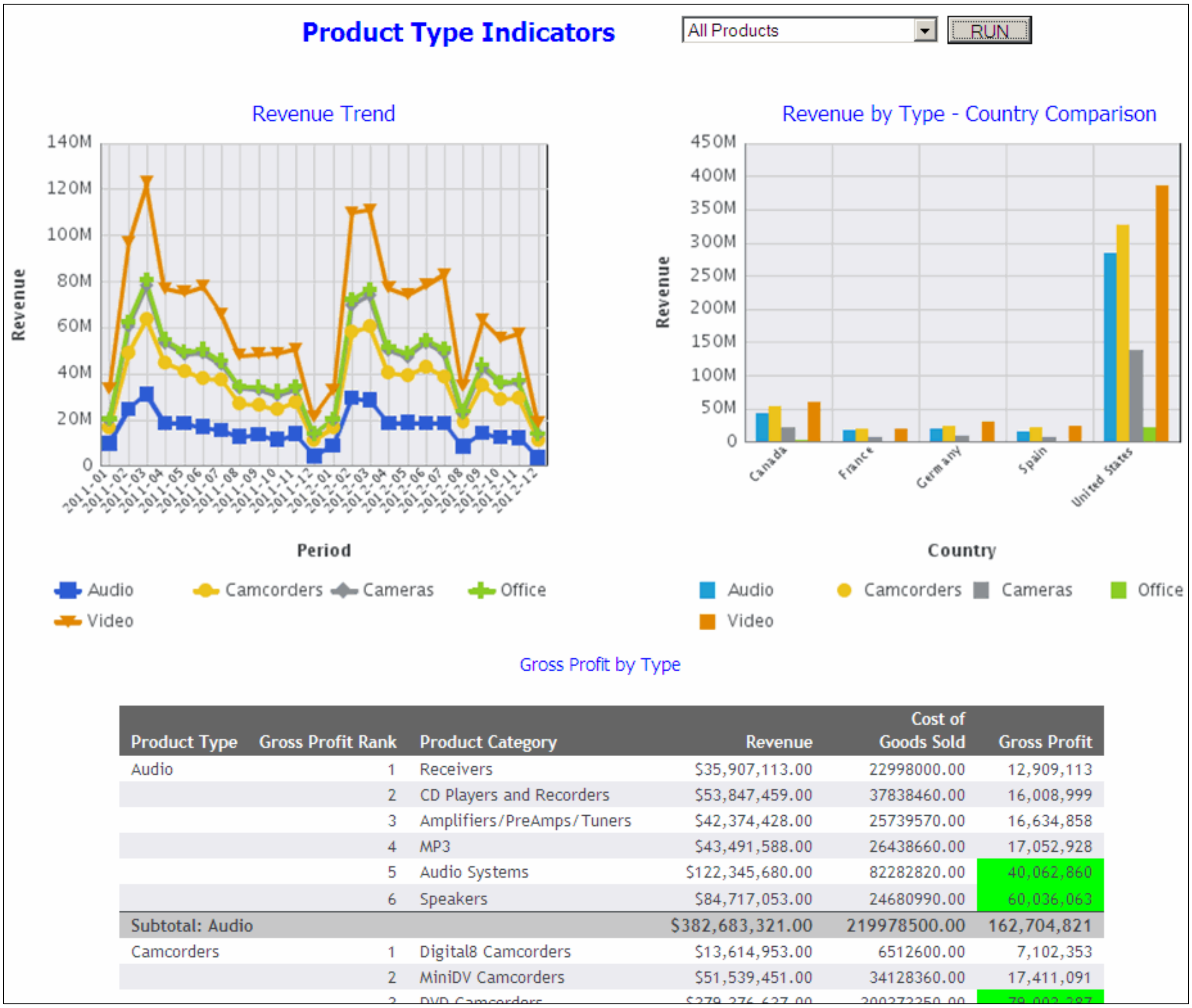
Figure 2-31 Coordinated chart

2.8.2 Developer Workbench HTML Composer

Reports/Charts/Graphs that were previously created in InfoAssist can be used in a dashboard that is by using the powerful Developer Workbench HTML Composer. This advanced dashboarding component handles parameters well because it uses various control types (calendar, drop-down lists, radio buttons, and so on). Advanced features such as chaining of parameters, or even adding Java Script is supported.

> **Note:** As of DB2 Web Query v2.2, Developer Workbench has a reworked interface that is enriched with additional functions, such as a mobile design feature.

Figure 2-32 shows a simple dashboard that is designed with Developer Workbench that incorporates two charts and one report that have the same filter set. The unique filter controller provides input for all the objects in the dashboard.



Product Type	Gross Profit Rank	Product Category	Revenue	Cost of Goods Sold	Gross Profit
Audio	1	Receivers	$35,907,113.00	22998000.00	12,909,113
	2	CD Players and Recorders	$53,847,459.00	37838460.00	16,008,999
	3	Amplifiers/PreAmps/Tuners	$42,374,428.00	25739570.00	16,634,858
	4	MP3	$43,491,588.00	26438660.00	17,052,928
	5	Audio Systems	$122,345,680.00	82282820.00	40,062,860
	6	Speakers	$84,717,053.00	24680990.00	60,036,063
Subtotal: Audio			$382,683,321.00	219978500.00	162,704,821
Camcorders	1	Digital8 Camcorders	$13,614,953.00	6512600.00	7,102,353
	2	MiniDV Camcorders	$51,539,451.00	34128360.00	17,411,091
	3	DVD Camcorders	$279,276,627.00	200273250.00	79,003,387

Figure 2-32 Compound report

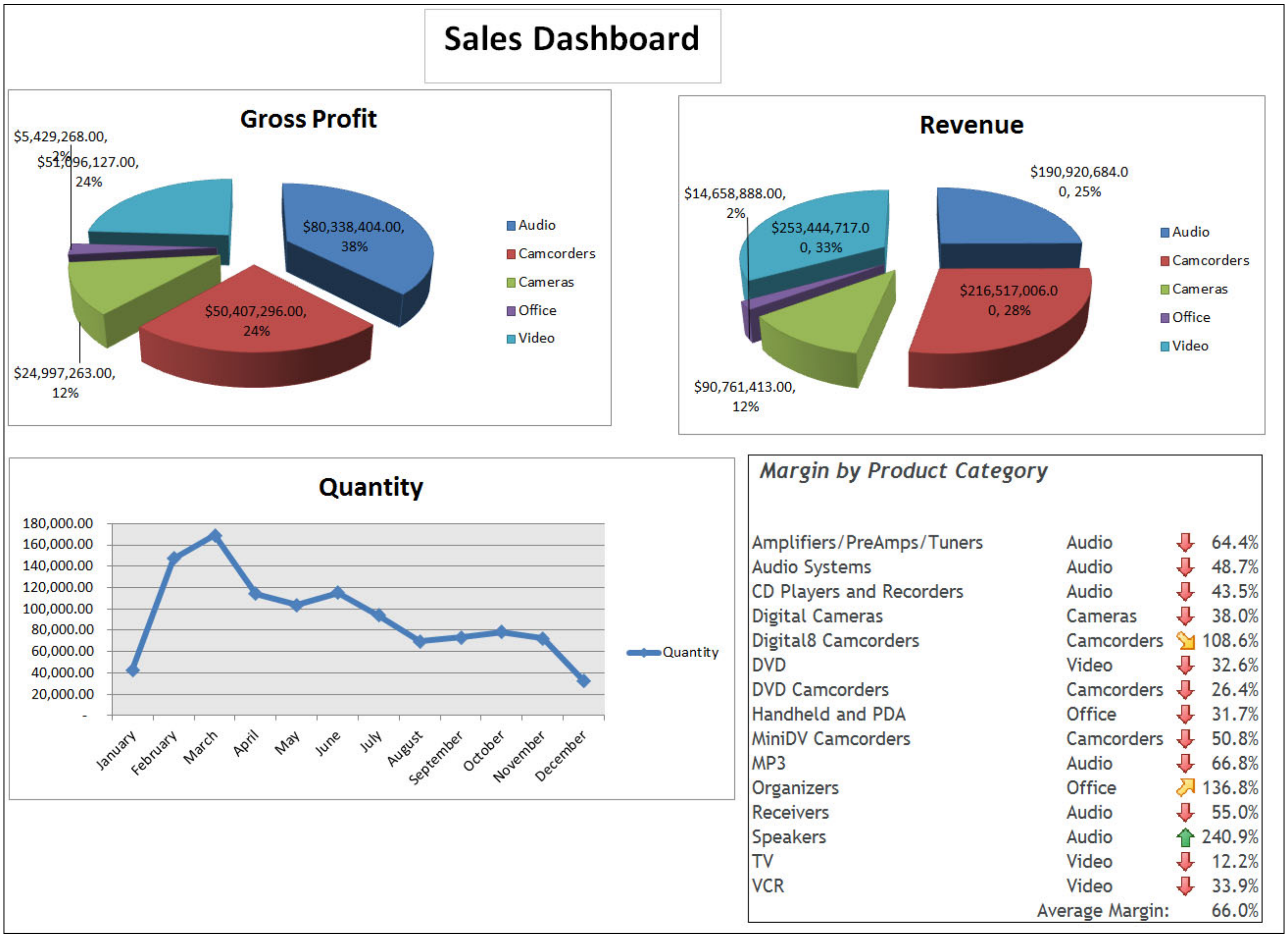
2.8.3 Spreadsheet Integration

Even with the growing popularity of Business Intelligence products such as DB2 Web Query, Microsoft Excel continues to be one of the most common programs that is used to generate reports and analyze data. Excel users continue to fall back to it when it comes to replacing Excel with a BI reporting tool, despite the challenges that are involved with accessing data within the Excel environment.

DB2 Web Query has responded to these demands and challenges with an Excel add-in provided with the product. Users can access and analyze all of their enterprise data without leaving their preferred spreadsheet world. The add-in allows Excel (2002 or higher) users to run DB2 Web Query reports from within Excel and populate an identified cell range. You can build sophisticated charts, scorecards, and dashboards by using Excel and use DB2 Web Query reports to populate the cells on which those components are based. Users do not need to go into the web browser to access the power of DB2 Web Query.

> **Note:** Installing the Excel add-In does requires some configuration. For more information, see the DB2 Web Query wiki.

An example of excel charts built over data returned to the spreadsheet through a DB2 Web Query report is shown in Figure 2-33.



Margin by Product Category

Amplifiers/PreAmps/Tuners	Audio	⬇	64.4%
Audio Systems	Audio	⬇	48.7%
CD Players and Recorders	Audio	⬇	43.5%
Digital Cameras	Cameras	⬇	38.0%
Digital8 Camcorders	Camcorders	↘	108.6%
DVD	Video	⬇	32.6%
DVD Camcorders	Camcorders	⬇	26.4%
Handheld and PDA	Office	⬇	31.7%
MiniDV Camcorders	Camcorders	⬇	50.8%
MP3	Audio	⬇	66.8%
Organizers	Office	↗	136.8%
Receivers	Audio	⬇	55.0%
Speakers	Audio	⬆	240.9%
TV	Video	⬇	12.2%
VCR	Video	⬇	33.9%
	Average Margin:		66.0%

Figure 2-33 Sales Dashboard that uses Excel add-in

2.9 Mobile

You can use DB2 Web Query to publish most objects for use with *mobile* devices. A list of the mobile browsers that were tested (and are fully supported) can be found in *DB2 Web Query New Features - Release 2.2.0*.

Publishing an object for mobile devices is a drag and drop action that is available to all users. Users connecting through a mobile device can access objects through standard browser features or the Mobile Faves app. Mobile Faves is a no-charge universal app for iOS and Android mobile devices that locally displays, stores, and manages any DB2 Web Query mobile content in all supported formats. You can also use the app to view and interact with dashboards.

Figure 2-34 shows a DB2 Web Query dashboard on a mobile device.

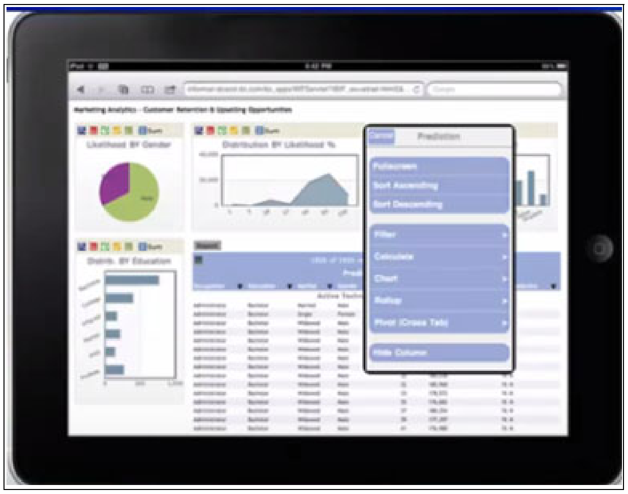


Figure 2-34 DB2 Web Query dashboard on a mobile device

2.10 Scheduling and distribution of reports

After you have your reports running, some users might be interested in having a specific set of information (reports and charts) to be delivered automatically to them periodically. For example, each sales representative wants to know how they are positioned towards their goals on weekly, monthly, or quarterly bases, and they want to receive this information by email instead of having to go to the BI portal and ask for the reports to be run. Another example is when you need to send information periodically to external people, for example you might need your agents to be informed about their customers' summaries but you do not want to give them direct access to your reporting system (no user profile is defined for them). In similar cases, it important to configure your BI solution with a scheduling and distribution solution.

DB2 Web Query Standard Edition scheduling functions offer you sophisticated scheduling and distribution capabilities:

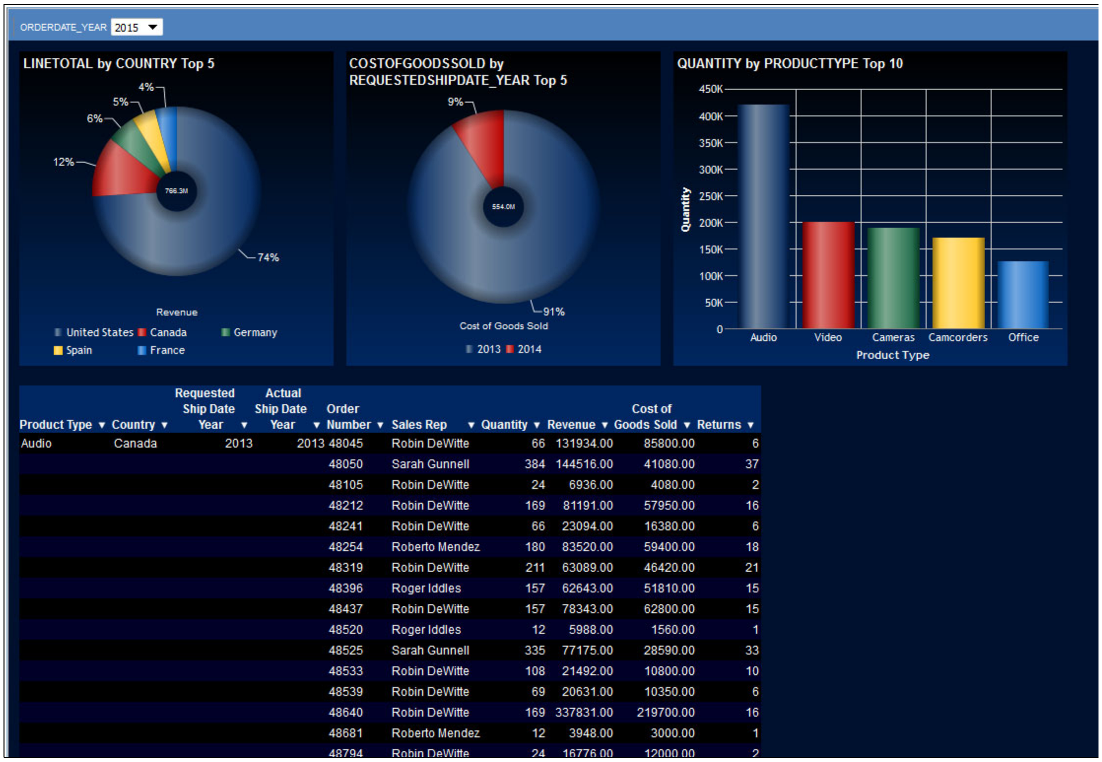
- Run reports at specified times (weekly, daily, M/W/F, exclude holidays)
- Deliver by email, ftp, or store in a network driver
- Targeting specific emails or a mailing list
- Bursting of aggregated reports for delivery of single section to a specific target
- Passing parameters to batch oriented report execution

2.11 Build Sample Reports with Wizards

With recent versions of DB2 Web Query (Express or Standard), two wizards are included to generate many sample reports over your data or uploaded spreadsheet data. Both wizards provide an easy way to "jump-start" your reporting environment by providing the steps that are needed to create synonyms (metadata) and autogenerate 20 - 30 objects (reports, charts, and dashboards). The actual number of objects varies based on your data (and how many measurement or numeric fields the wizard finds). True date fields are also automatically decomposed into date attributes such as month, quarter, and year, and the wizards attempt to determine hierarchical relationships in fields that represent a dimension. A dimension might be "geography" where you have fields such as COUNTRY, STATE, and CITY. The wizard builds a relationship for these fields to enable reports such as the auto drill-down report type.

The two wizards are the *metadata wizard* and the *upload wizard*. The metadata wizard builds synonyms and reports over your DB2 for i data. The upload wizard does a similar task, but for spreadsheet or flat file data that is imported to DB2 for i as relational data and then has the metadata and the pertinent reporting added.

Figure 2-35 shows one of the basic dashboards that is by the wizard on the sample database data.



Product Type ▾	Country ▾	Requested Ship Date Year ▾	Actual Ship Date Year ▾	Order Number ▾	Sales Rep ▾	Quantity ▾	Revenue ▾	Cost of Goods Sold ▾	Returns ▾
Audio	Canada	2013	2013	48045	Robin DeWitte	66	131934.00	85800.00	6
				48050	Sarah Gunnell	384	144516.00	41080.00	37
				48105	Robin DeWitte	24	6936.00	4080.00	2
				48212	Robin DeWitte	169	81191.00	57950.00	16
				48241	Robin DeWitte	66	23094.00	16380.00	6
				48254	Roberto Mendez	180	83520.00	59400.00	18
				48319	Robin DeWitte	211	63089.00	46420.00	21
				48396	Roger Iddles	157	62643.00	51810.00	15
				48437	Robin DeWitte	157	78343.00	62800.00	15
				48520	Roger Iddles	12	5988.00	1560.00	1
				48525	Sarah Gunnell	335	77175.00	28590.00	33
				48533	Robin DeWitte	108	21492.00	10800.00	10
				48539	Robin DeWitte	69	20631.00	10350.00	6
				48640	Robin DeWitte	169	337831.00	219700.00	16
				48681	Roberto Mendez	12	3948.00	3000.00	1
				48794	Robin DeWitte	24	16776.00	12000.00	2

Figure 2-35 Wizard-created dashboard.

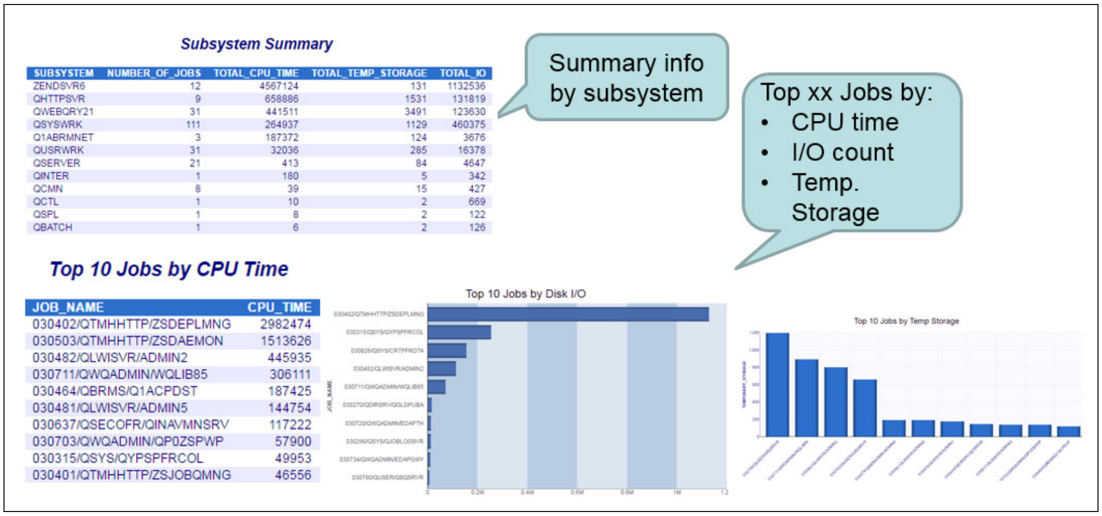
All generated objects can be edited if some customization is needed, or you can use them as templates for further development.

2.12 Building Reports over IBM i Services

In addition to querying your production database and business data and transforming that data into information, DB2 Web Query can provide information about the database itself. As examples, you might want to track who is allocating disk space, who is the biggest spool file creator, which jobs are the most resource-intensive, which tables have the most deleted rows, and which database tables are close to the system limit.

Figure 2-36 shows an example of a DB2 Web Query report built over a DB2 for i service QSYS2.ACTIVE_JOB_INFO. It gathers and presents information about the current subsystems metrics (number of jobs, CPU time, total temporary storage, and I/O operations) together with a summary of top 10 jobs by CPU, I/O, and temporary storage.

> **Note:** Compare the amount of work that is needed to gather, arrange, and present this information by using traditional methods (5250 screen, control language (CL) commands, output to spool files, and so on) with the simple techniques that are presented in this section. Also, you can get this information as a dashboard on your mobile device or have delivered through email at scheduled times or when a threshold is hit.



Subsystem Summary

SUBSYSTEM	NUMBER_OF_JOBS	TOTAL_CPU_TIME	TOTAL_TEMP_STORAGE	TOTAL_IO
ZENDSVR6	12	4567124	131	1132536
QHTTPSVR	9	658886	1531	131819
QWEBQRY21	31	441511	3491	123630
QSYSWRK	111	264937	1129	460375
Q1ABRMNET	3	187372	124	3676
QUSRWRK	31	32036	285	16378
QSERVER	21	413	84	4647
QINTER	1	180	5	342
QCMN	8	39	15	427
QCTL	1	10	2	669
QSPL	1	8	2	122
QBATCH	1	6	2	126

Summary info by subsystem

Top xx Jobs by:
- CPU time
- I/O count
- Temp. Storage

Top 10 Jobs by CPU Time

JOB_NAME	CPU_TIME
030402/QTMHHTTP/ZSDEPLMNG	2982474
030503/QTMHHTTP/ZSDAEMON	1513626
030482/QLWISVR/ADMIN2	445935
030711/QWQADMIN/WQLIB85	306111
030464/QBRMS/Q1ACPDST	187425
030481/QLWISVR/ADMIN5	144754
030637/QSECOFR/QINAVMNSRV	117222
030703/QWQADMIN/QP0ZSPWP	57900
030315/QSYS/QYPSPFRCOL	49953
030401/QTMHHTTP/ZSJOBQMNG	46556

Figure 2-36 Usage of QSYS2.ACTIVE_JOB_INFO service

If you're a Systems Administrator, DB2 Web Query can modernize your traditional way of monitoring and tracking utilization metrics. IBM is providing a number of sample System Administrator type reports at no charge - you can request them by sending an email to QU2@us.ibm.com. One sample dashboard covers spool file usage, as depicted in Figure 2-37.

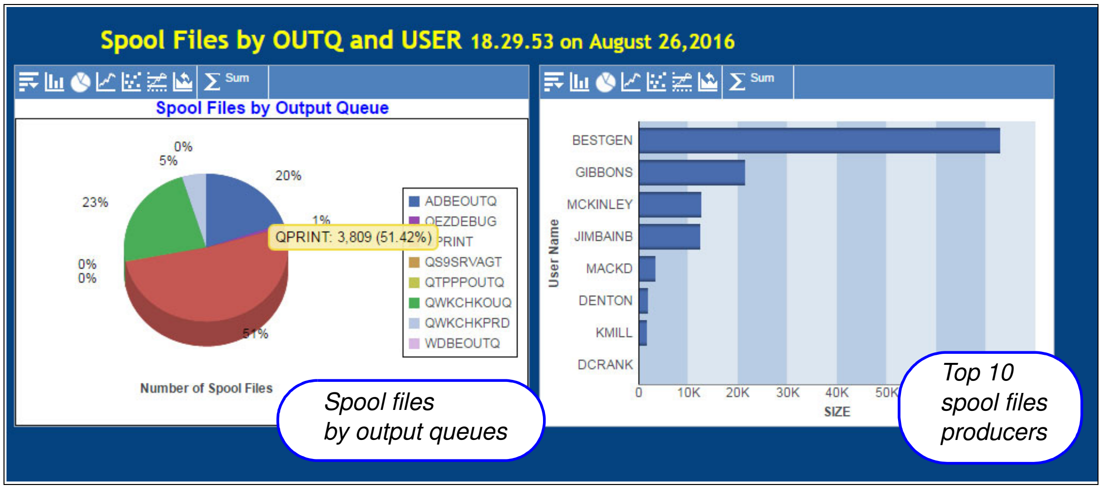


Figure 2-37 Usage of QSYS2.OUTPUT_QUEUE_ENTRIES service

These IBM i services are SQL versions of commands previously only available through proprietary APIs and system low-level interfaces. The services include both database and operating system functions and offer the following advantages:

- They can be used as building blocks for additional SQL programming.
- They can reduce the need for CL programming by allowing access to services that were only previously available by using CL commands.
- They can reduce the need for writing high-level language (HLL) programs to manage cumbersome calls to APIs that are provided by IBM.

For more information about DB2 for i Services and IBM i Services, see IBM Knowledge Center.

The DB2 for i Wiki also provides additional services documentation.

Most of these services are available in IBM i 7.1 or later, although over time only newer releases will have additional services provided. Many services are implemented as part of a Technology Refresh (TR), which means that they are supported after the installation of a certain program temporary fix (PTF) group. Some services, for example temporary storage usage, are available in IBM i 7.3 and IBM i 7.2 only.

IBM provides these services by using procedures, user-defined functions (UDFs), and views, which sometimes conceal the use of user-defined table functions (UDTFs). They are shipped in either the QSYS2 or SYSTOOLS library. Library QSYS2 contains the services that are considered part of the operating system, and they provide upward compatibility.

Library SYSTOOLS is used by IBM to provide useful functions and examples about how to use the power of SQL. By examining the underlying SQL code, you can gain insight about the techniques that are used and get inspiration for further development. The examples in SYSTOOLS are provided "as is" with no guarantee of upward compatibility.

DB2 for i Services covers various areas:

- Application
- Performance
- Plan cache
- Utility

IBM i Services cover various functional areas in the operating system:

- Application
- Java
- Job
- Journal
- Librarian
- Message handling
- Product
- PTF
- Security
- Spool
- Storage
- System health
- Communications
- Work management

It is out of the scope of this book to provide details about how to use DB2 for i and IBM i Services. But you can request sample services and documentation on how these are built by sending an email to QU2@us.ibm.com

Read more about DB2 Web Query EZ-Install and System Admin services at:

http://Ibm.biz/db2wqezinstall-info

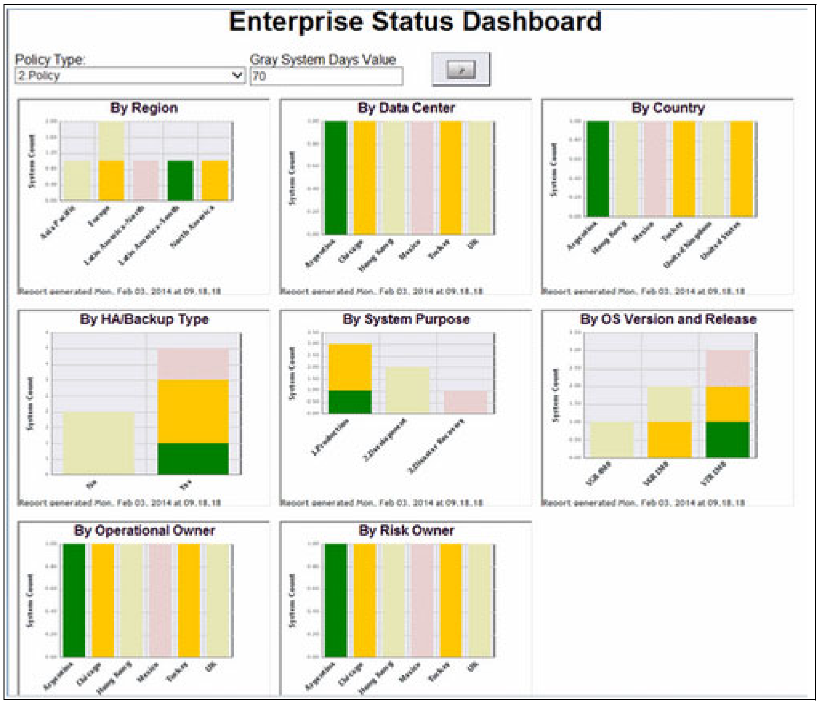
2.13 IBM i Compliance and Reporting Tool

Security is one of the main concerns for system administrators and top-level management in general. Many liabilities can arise from a security breach, and the consequences can be expensive in terms of money and customer good will. Systems must comply to given sets of rules and must be monitored to maintain compliance. Without monitoring, the state of the system is unknown, and although you might have been secure at one point in time, without ongoing monitoring, you are not sure what your status is.

IBM developers created a tool specifically for IBM i that takes advantage of the unique features of the system, which includes DB2 Web Query as the best tool for presenting and distributing information.

The IBM i Compliance and Reporting Tool (CART) provides evidence that risk is being managed according to enterprise-defined risk thresholds, which empowers Senior management to make informed risk management decisions about where best to allocate resources.

CART provides a centralized view of the security compliance status across the enterprise with no access to remote machines required, maintaining segregation of duties. This is true management visibility with meaningful reports that drive action. An example of the dashboards is provided in Figure 2-38.



Enterprise Status Dashboard

Policy Type: 2 Policy

Gray System Days Value: 70

Figure 2-38 IBM i Compliance and Reporting Tool: Enterprise Status Dashboard

Although CART provides many built-in tests and monitors, it is customizable to your applications and needs. CART provides measurable results, the ability to define Key Risk Indicators (KRIs), and traceability back to Security Standards and Company Policies. All of this information is available through a text interface, DB2 Web Query Dashboards, or your favorite reporting tool because all results are stored in an industry-standard SQL data warehouse.

Here is a list of some of these tests and monitors:

- Profile Analysis:
  - Special Authorities / Inherited Privileges
  - Group Profiles / Ambiguous Profiles
  - Default Passwords / Password Expiration
  - Inactive Accounts
  - *PUBLICly Authorized Profiles
  - Privately Authorized Profiles
  - Initial Programs, Menus, and Attention Programs
  - Command Line Access

- Administration and Configuration:
  - System Values / Audit Control Settings
  - Invalid Signon attempts
  - Work Management Analysis
  - Service Tools (SST) Security
  - DDM Password Requirements
  - Registered Exit Points / Exit Programs
  - Function Usage
  - Library Analysis / *ALLOBJ Inheritance
  - PTF Currency
  - Customer Defined Events and Items
  - CPU/DASD Utilization and Availability
  - Actionable Security Events in near real time!
- Network Settings:
  - Network attributes / Time Server
  - NetServer Configuration
  - TCP/IP servers / Autostart values
  - Digital Certificate Expiration
  - SNMP / SSH / SSL Configuration
  - Listening ports / Network Encryption
  - IP Datagram Forwarding
  - IP Source Routing
  - APPN Configuration

CART is available from IBM Lab Services. For more information, go to this website.

3

Installation and server operations

This chapter explains the installation and operation of DB2 Web Query, product 5733-WQX. Because installation and PTF requirements can be a somewhat dynamic set of instructions/procedures, the installation and setup instructions in this chapter are purposely high level and you should refer instead to the EZ-Install package or the DB2 Web Query INSTALLATION link off the wiki: ibm.co/db2wqwiki.

Note: Since the original draft mode of this document was completed, IBM has come out with a DB2 Web Query EZ-Install package which is a simplified installation and setup procedure. To read more about it and learn how to request it, you can go to http://ibm.biz/db2wqezinstall-info. For most customers, EZ-Install should replace the previously (and still) available traditional method.

3.1 Installation and setup

The DB2 Web Query for i licensed program product number for installing is 5733-WQX. There are several ORDERING product numbers, including:

* 5733-WQe for Express Edition

* 5733-WQs for Standard Edition

* 5733-WQm for DataMigrator ETL

3.1.1 Installing DB2 Web Query for i

Important: Note again that installing and setting up DB2 Web Query is much simpler with the latest DB2 Web Query EZ-Install. However, if for some reason you would like to install in the traditional manner, those instructions can be found at the DB2 Web Query wiki at ibm.co/db2wqwiki. Take the INSTALLATION link and carefully review the INSTALL GUIDE and INFO APAR documents for the latest installation instructions, pre-req, and PTF information.

Rather than specify EZ-Install instructions here in this document, it is best to get the latest installation instructions upon requesting the EZ-Install package. EZ-Install will install ALL features and products of DB2 Web Query, as well as some other goodies, including:

1. Both Express and Standard Edition, with Standard enabled by default
2. The latest PTF Group available for DB2 Web Query
3. Developer Workbench Windows Client installation file
4. DB2 Web Query DataMigrator ETL tool for building data marts or data warehouses
5. A SAMPLE database containing data used in the self guide tutorials and also including a date dimension table and several views and stored procedures.
6. A SAMPLE set of reports for the IBM i Systems Administrator
7. A completed set of tutorial reports that can also server as sample reports
8. A Query/400 Discovery tool

Important: In general, a new Group PTF for Web Query is released about every three to four months. It is best to try to stay current with these updates. If you run into a problem and call IBM Support, you will be asked whether you are at the most current Group PTF for DB2 Web Query. The latest information about Group PTFs can be found at this website.

The EZ-Install package also includes a pre-req checker you can run first to determine if you’re missing or downlevel on any pre-reqs before doing the full install. The package also makes available the Developer Workbench client installation executable.

IBM i system objects

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

- `/QIBM/PRODDATA/QWEBQRY`
- `/QIBM/USERDATA/QWEBQRY`

Table 3-1 lists the system objects that are created. The libraries that are listed contain additional objects that are not shown.

Table 3-1 System objects

Object name	Object type
QWEBQRYX	*LIB
QWEBQRY	*LIB
QWQREPOS	*LIB
QWQCENT	*LIB
QSYS/QWQADMIN	*USRPRF
QSYS/QWQ0000000	*AUTL
QUSRSYS/QWQADMIN	*MSGQ

DB2 Web Query also creates and uses one user profile for the product that is called *QWQADMIN*. This profile has functions on the server side, has the authority to start and stop the Reporting Server, and is used to initially assign Web Query administrators. The profile is automatically created during installation.

EZ-Install will allow you to specify a non-Qxxxx user profile to be a DB2 Web Query Administrator, will automatically add that user profile as a DB2 Web Query user, and assign them as an administrator within Security Center.

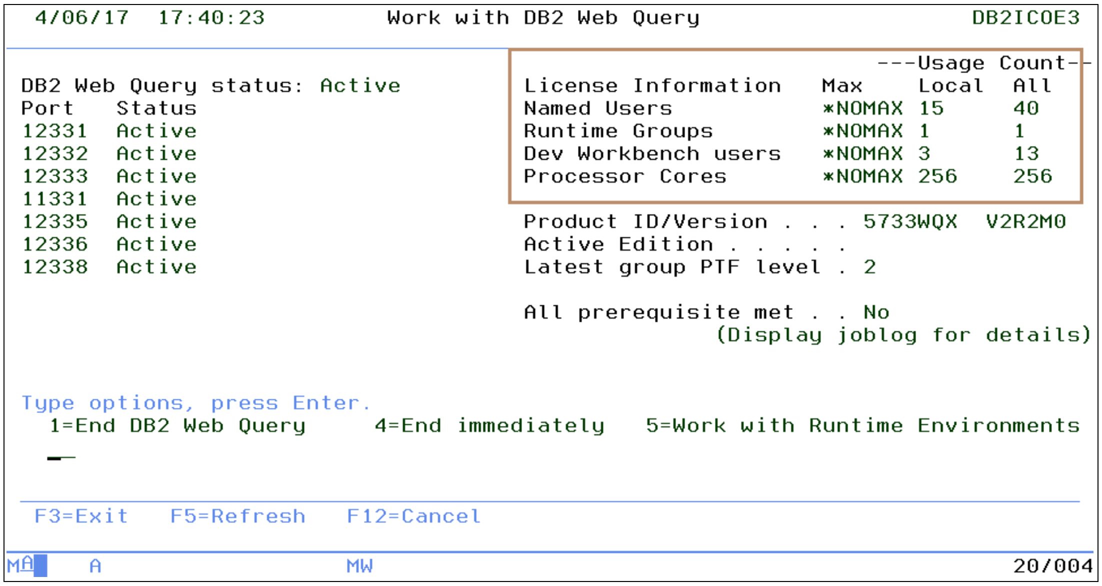
3.1.2 Authorizing and verifying users

Users must be licensed to use DB2 Web Query. There are multiple types of licenses that are available, depending on usage intent.

The different types of user licenses include Authorized Licensed Users, Developer Workbench Users, and Runtime Groups. Every DB2 Web Query for i user must be authorized to be able to sign into DB2 Web Query, either directly or as assigned to a group profile that itself is assigned to the product.

To add or remove licenses from user profiles, use the Security Center. For more information about the various licenses and the use of the Security Center, see Chapter 6, “IBM DB2 Web Query for i Security Center” on page 127.

You can see the status of the licensed users by running the **WRKWEBQRY** command and viewing the usage count information, as shown in Figure 3-1.



```
 4/06/17  17:40:23         Work with DB2 Web Query                     DB2ICOE3

                                                            ---Usage Count--
DB2 Web Query status: Active           License Information   Max    Local  All
Port   Status                          Named Users           *NOMAX 15     40
12331  Active                          Runtime Groups        *NOMAX 1      1
12332  Active                          Dev Workbench users   *NOMAX 3      13
12333  Active                          Processor Cores       *NOMAX 256    256
11331  Active
12335  Active                          Product ID/Version . . . 5733WQX  V2R2M0
12336  Active                          Active Edition . . . . .
12338  Active                          Latest group PTF level . 2

                                       All prerequisite met . . No
                                                     (Display joblog for details)


Type options, press Enter.
  1=End DB2 Web Query      4=End immediately    5=Work with Runtime Environments
  _

 F3=Exit    F5=Refresh    F12=Cancel

MA     A                  MW                                                20/004
```

Figure 3-1   License information

3.1.3 License keys

After installing DB2 Web Query for i with EZ-Install, you are allowed a 70-day trial period in which to try the products. The grace period starts the first time that DB2 Web Query is started.

During this grace period, you have access to all the features and products, with unlimited users as well. After the grace period expires, license keys are required to continue using DB2 Web Query and you’ll be restricted to using what your licenses dictate.

To obtain DB2 Web Query license keys, work through your normal product ordering channels (for example, an IBM Business Partner). The system serial number is required to generate the appropriate license keys.

A license key is needed to activate the base product feature (5050), either Express Edition (5101) or Standard Edition (5102) feature, and licensed users (5104) feature. Depending on your needs, you might also need user licenses for the Developer Workbench (5105) feature or Runtime Groups (5106) feature. If you purchase companion features or products such as the JDE adapter (5107) or DataMigrator (5108), you must apply those license keys as well.

To enter your license keys and activate the specific features, run the **ADDLICKEY** command.

Note: It is not necessary to wait for the grace period to expire before adding license keys. Keys can be added at any time. However, once added, you are restricted to products and user licenses as dictated by your keys.

3.1.4 Dynamic Language Switching

If you are running a system that might need to be viewed in multiple languages, the Dynamic Language Switching option is for you. It is included in the base product. You can use this option to set a list of languages that a user can select from at login time. This feature, by default, updates the interface of the program to match the language that is selected. Figure 3-2 shows more details.

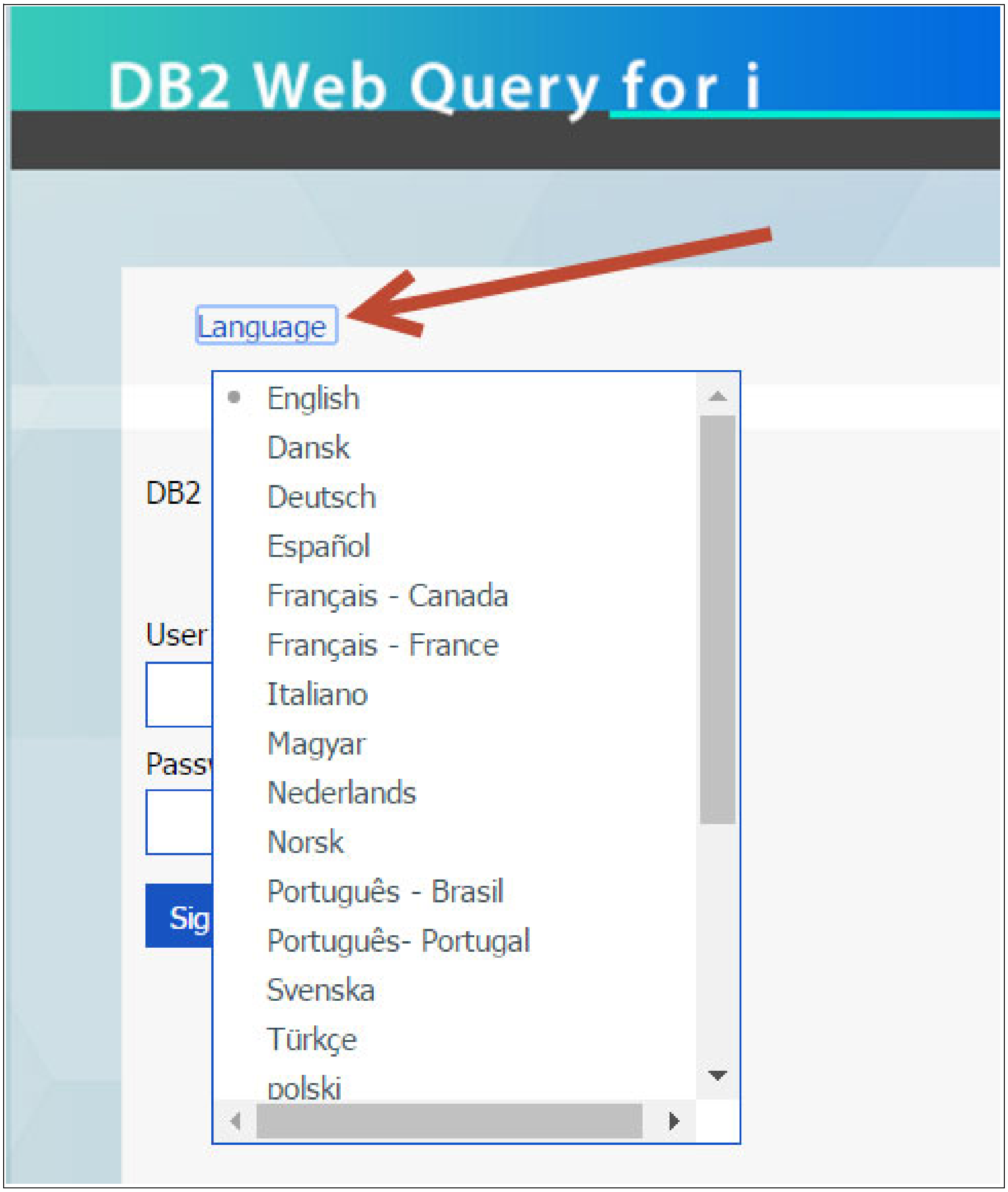
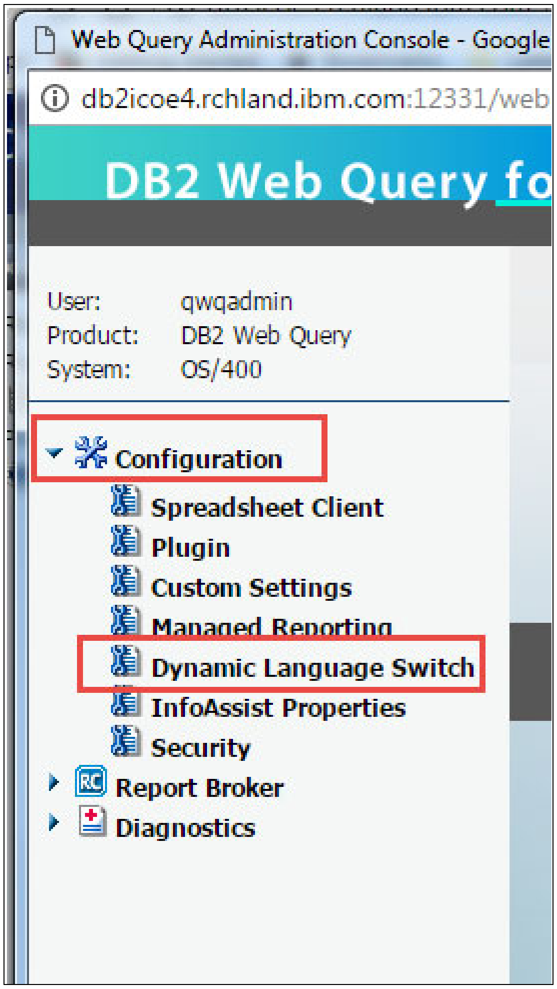


Figure 3-2 Language selection

To enable this feature, sign in as user QWQADMIN. From the ADMINISTRATION link, open up the ADMINISTRATION CONSOLE. Expand the CONFIGURATION section and select Dynamic Langauge Switch. Choose which languages to make available to end uers and SAVE.



Web Query Administration Console - Google

db2icoe4.rchland.ibm.com:12331/web

DB2 Web Query fo

User: qwqadmin
Product: DB2 Web Query
System: OS/400

For more information, see instructions in the installation document on the DB2 Web Query wiki at this website.

3.1.5 Sample database

EZ-Install installs a sample set of tables, views, and stored procedures that are used in a set of tutorials that you can use to learn how to build reports with the InfoAssist component. The tutorials are available from the DB2 Web Query wiki at ibm.co/db2wqwiki.

3.2 Requirements

This section covers the IBM i and PC minimum requirements to install and run DB2 Web Query for i.

3.2.1 IBM i requirements

Your IBM i environment must be at IBM i OS level 7.1 or later. As of this writing, you need to be at a minimum of Version 2.2 for OS level 7.3. A general guideline, or minimums, for resource requirements is 3000 CPW, 8GB of memory and 16 disk arms. However, this is very subjective and many factors can affect performance.

3.2.2 Web browser requirements

The list of supported browsers is quite dynamic due to the nature of browser changes. Please see the New Features document that is appropriate for your DB2 Web Query release at this website under the DOCUMENTATION link.

3.2.3 PC requirements for Developer Workbench

Although not required for DB2 Web Query report development and execution, many people find that the Developer Workbench client provides additional development features and ease of use that make it very worthwhile. If you choose to install Developer Workbench, a Windows based PC is required.

Hardware guidelines

Although they are not strict requirements, the following minimums are guidelines to achieve acceptable performance:

- 2 GHz or faster dual core 32-bit (x86) or 64-bit (x64) processor
- 4 GB of RAM minimum
- 3.5 GB of free disk space for download and installation

Software requirements

The Windows machine must meet the following minimum software requirements:

- Windows 7 or higher or Windows 2008 R2 or higher Server edition.
- Developer Workbench launches Microsoft Internet Explorer for certain features. See the supported web browser list for supported Internet Explorer browser levels.

For more information, consult the DB2 Web Query for i wiki at this website.

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

3.3 Administrative commands

DB2 Web Query provides several IBM i CL commands to help with administrative tasks. By default, some of these commands are shipped as *PUBLIC *EXCLUDE security setting. However, any user that is a DB2 Web Query administrator (a member of the QWQADMIN group profile) or is authorized to use the commands through the *ALLOBJ special authority or explicit authority can use these commands.

When DB2 Web Query is started, a prerequisite check is performed. If required products or PTFs are missing, DB2 Web Query will not start. The job log provides details about what is missing so that you can correct the issue.

Here are the most commonly used CL commands:

`WRKWEBQRY` This is the main DB2 Web Query command. The **`WRKWEBQRY`** command can be used to start, stop, and display the status of DB2 Web Query. Each Web Query port and its status is displayed, installed product and prerequisite information is shown, license information is provided, and the prerequisite status is given. Additional capabilities such as working with runtime environments is also provided.

STRWEBQRY 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.

WRKWQRTE Use the **WRKWQRTE** command to create dynamic Runtime Environments and assign users to them. After they are created and assigned, users can quickly and easily change their library list before running reports. For more information, see the Dynamic Runtime Environments document at this website.

CRTWQSYN Use the **CRTWQSYN** command to create and refresh DB2 Web Query synonyms. It is a CL alternative to the main metadata console, which is available from the browser, and the Developer Workbench interface to create synonyms.

RUNWQFEX Use the **RUNWQFEX** command to run Web Query reports from the IBM i command line. Using this command requires Standard Edition.

3.4 Subsystem and jobs

DB2 Web Query runs in its own IBM i subsystem, QWEBQRY21. All product-related jobs and associated database jobs running on the product's behalf run within this subsystem.

Typically, DB2 Web Query uses the SQL native call-level interface (CLI) to access the DB2 for i database. CLI is a callable SQL programming interface that is built into DB2 for i. CLI consists of application programming interfaces (APIs) that are used to connect to the server and run dynamic SQL statements. CLI is a subset of Open Database Connectivity (ODBC). The database jobs that processes CLI are named QSQSRVR. Whenever a QSQSRVR job is active in the QWEBQRY21 subsystem, it is performing database work on behalf of DB2 Web Query for i.

When DB2 Web Query is active, the following jobs run in the QWEBQRY21 subsystem:

- EDAPTH: A single job that helps manage the workspace process. It runs under user profile QWQADMIN.
- EDAPLOG: A single job that contains startup information. It runs under user profile QWQADMIN.
- EDAPGWY: Three listener jobs, one each for HTTP, TCP, and Java. These jobs receive incoming requests and hand off work to the TSCOM3 jobs. All the jobs are under user profile QWQADMIN.
- JSCOM3: These are the main worker jobs. When a report is run, these jobs accept the request from the EDAPGWY job and translate the DB2 Web Query for i request into SQL. One TSCOM3 job does not necessarily correspond to one user. A single TSCOM3 job can service several users, contributing to the scalability of the product. These jobs are referred to as *agents*, and run under (swap to) the current user's profile.
- JSCOM3C: A single job that services Java processes. It runs under user profile QWQADMIN.
- WQLIB85: These jobs are used as integrated application server jobs. Four of these jobs should be running whenever DB2 Web Query for i is active. Three jobs run under user profile QTMHHTTP, the other under profile QWQADMIN.

- FDS: This single job is used for internal log writing.
- QSQSRVR: This database prestart job handles SQL requests that are made over CLI. There can be several of these QSQSRVR jobs active at once, depending on what work is being requested from DB2 Web Query for i. These jobs do the actual database work of optimization, execution, and returning the results. By default, these jobs run under the user profile QUSER, but when a report is run, a CLI connect event occurs with one of these jobs. The user profile running the job is swapped to the user profile that initiated the report request.

In addition, the following jobs are started to support report scheduling and distribution features.

- STRBROKER: A single job that runs under user profile QWQADMIN and invokes QSHELL to run a script to start the Distribution Server.
- QP0ZSPWP: Two jobs that run under either user profile QWQADMIN or QSECOFR. One job is the JVM and the other is the console window that the Distribution Server starts.

3.5 Port usage

When DB2 Web Query is running properly, it is active on ports 12331 - 12339, not including port 12337. Port 12339 is used for Report Broker (report scheduler), which is part of Standard Edition. If Standard Edition is not installed or not licensed (and no longer in grace period), port 12339 is not active. In addition, if DB2 Web Query is in trial, port 12334 is active. If DB2 Web Query is licensed, port 11331 is active instead of 12334.

The **WRKWEBQRY** command is normally the best way to display the status of these ports. Alternatively, you can look at active ports by using the **NETSTAT** command and using option 3. After the display is up, select F14 to display port numbers. As you scroll down, you see that the local port column is ordered descending. Scroll down until you reach a local port of 12331 to see these connections.

3.6 Running at subcapacity

DB2 Web Query is defined as a core-based product, which means that the product is purchased and licensed based on the number of (hardware) cores it is allowed to use. A primary advantage of this approach is that customers can buy what they need. They can purchase fewer cores of DB2 Web Query than they have active on their IBM i system, For example, a customer running with a two core IBM i partition can purchase and install just one core of DB2 Web Query. This concept is called *subcapacity licensing*.

To support this feature, the product automatically enforces this core licensing when it runs to avoid any potential violation of the license agreement. This feature greatly simplifies management of the product and avoids any potential violation of the license.

An advantage is that customers can be assured that DB2 Web Query and its associated database work do not consume more resources than the license allows. A potential disadvantage is that the number of licensed cores might limit its performance and responsiveness, and simply adjusting the capacity of the partition does not necessarily improve responsiveness. As part of capacity planning, it is important to consider the number of licensed cores that are purchased for DB2 Web Query.

3.7 Running QU2 and WQX concurrently

Customers that migrate from Web Query V1.1.x (product ID 5733QU2 or QU2) to Version 2.2 (5733WQX - WQX) can run both versions concurrently under the following conditions:

- A permanent WQX license key is *not* applied.
- The trial period for WQX has not ended. The trial period initiates the first time WQX is started by using either the **STRWEBQRY** or **WRKEWBQRY** command, or when EZ-install is used to install and start the product. It ends 70 days after it is initiated.

When the WQX trial period is over, permanent license keys must be entered to continue using the product. After this occurs, the customer cannot start both QU2 and WQX at the same time. They can start one or the other, but not both. If one is active, the other cannot start.

A migration utility is included in DB2 Web Query (as of Version 1.1.2) that can migrate your Version 1 (QU2) reports and meta data (and other reporting objects) into Version 2. For more details about the Migration utility, refer to the INSTALLATION link of the DB2 Web Query wiki at ibm.co/db2wqwiki.

4

Defining metadata

This chapter explains what metadata is and why you should consider it as an advantageous feature, something that can reduce the complexity of your query and reporting environment and make life easier for your report developers.While you can use metadata minimally, the benefits of ehancing it over time are many, incuding:

- Defining business rules for virtual fields in one place - single version of the truth
- Eliminating complex data untangling for report authors
- Simplified use of date fields in reports
- Leveraging existing procedures, views, and functions
- Documenting your database
- Significantly reduced mantenance by shielding reports when underlying data changes

Before you can create a single report or graph in IBM DB2 Web Query for i, you must first create metadata (known as *synonyms* in DB2 Web Query terms) for the data source. The metadata requirement of the DB2 Web Query for i product might be a new concept to you, but is a standard practice for leading Business Intelligenc solutions in the industry. Typical questions from those not familiar with the concept could include:

- What exactly is metadata?
- Why do you need it?
- Can you not query your files directly?
- What happens if the structure of my underlying files changes?

This chapter provides an overview of the metadata topic and why metadata is an important element of DB2 Web Query.

4.1 What is metadata

Metadata in its simplest form is just data about data. Whenever you issue the **DSPFD** or **DSPFFD** command, what is generated and displayed on your screen is in fact metadata, which is information such as record lengths, record formats, field names, data types, field attributes, and field lengths. DB2 for i also maintains system catalogs, which are files that store information about each of the objects in your database. Catalogs are effectively a materialized metadata repository that is kept up to date and can be queried to collect a wide variety of information about your database.

In much the same way, DB2 Web Query metadata is a materialized repository that contains information about your database files. DB2 Web Query has the potential to read all of your database information, but you must first give it information (metadata) about where the data is and how it is structured. Before you can create a report or graph, you must first create metadata (also referred to as a synonym) over the data source. Think of SYNONYMS as a metadata object in DB2 Web Query. It contains no data from your files/tables, but represents that data.

You can create a synonym over the following database and other object types:

- Tables/physical files
- SQL views
- DDS logical files (but this is generally discouraged)
- Stored procedures
- External SQL scripts
- Table log (journal) records
- Materialized query tables
- Structured text files, such as comma-separated variable (`.csv`) or Microsoft Excel (`.xlsx`) by using the Upload function

In general, whenever you create a synonym over an object, database, or flat file, two text files are created in the Integrated File System (IFS) in a directory that has the same name as your application folder in DB2 Web Query for i. Application folders are where synonyms reside from the perspective of a developer using the GUI interfaces. When you create a “Top Level Folder” for your reporting objects, DB2 Web Query will also create an “application” folder of the same name. There is also a global application folder called baseapp. If you create synonyms in that folder, report authors authorized to build reports in ANY top level folder can see those synonyms and effectively build reports over the data they represent.

```
/qibm/UserData/qwebqry/apps/<myfolder>
```

The reality is you’ll not really need to know about these files very often, or care what the structure is, as the recommendation would be to work with synonyms through the GUI user interfaces and rather than go into and edit these text files. Having said that, there are times when a master file or an access file is referenced in the software or documentation.

Synonyms can be created either in the shared application folder `baseapp` where they can be used by every report author, or they can be created in a private application folder that is associated with a specific top-level folder of reports. Top-level folders are typically organized based on either the report subject area (such as Manufacturing reports, Billing reports, Customer Service reports, or Sales reports) or on who is using the reports (for example, the Financial reports for the CEO/President/Business Analyst, salesperson reports for the Sales Managers, or personnel reports for the HR staff).

Separating synonyms into one or more application folders can thus serve two main purposes:

- Security: Synonyms that are associated to an application folder can be modified only by those users that have DBA privileges to the specific folder.
- Privacy: Synonyms that are associated to an application folder are presented only to those users that can develop in the specific folder and are only usable by those users that can use the folder and its contents.

An additional advantage of metadata is that it provides an abstraction layer between your data sources and those users that use the data in their reports. The data administrator can standardize the access to the data, the representation of the data on all reports (titles, formatting), and any calculations or business logic that are needed when creating calculated fields. For example, you can build business logic into a synonym so that all reports and graphs that reference those synonyms have access to all of that logic. Placing this business logic into the metadata provides standardization because all report developers use this same field whenever it is needed on a report. This concept supports the notion of a single version of the truth in all of your reports because each report developer does not have to re-create the business logic for every report. Standardizing of business logic in the metadata also helps eliminate reporting errors because everyone is using the trusted source of the metadata instead of having to re-create the business logic for each report (reducing errors). Proper standardization and simplification of your metadata greatly simplifies the environment for your report developers.

4.1.1 Benefits of metadata

The primary objective of the metadata layer is simple: Improve the productivity of the report developers by providing an abstraction layer and hiding database complexity. Report developers can be more productive if you keep the data model simple.

A simple and intuitive data model also means that you can extend the report developer community because developing new reports no longer requires an intimate knowledge of the data (such as the tables that are involved and the join conditions). By empowering more of your users, you can reduce their dependence on IT and the number of backlogged reporting requests.

But before this can happen, some work needs to be done to create and enhance this abstraction layer by using the tools that are provided with DB2 Web Query.

4.2 Creating synonyms

DB2 Web Query metadata (synonyms) can be created or sourced in several ways:

- DB2 Web Query provides several interfaces that you can use to create or edit synonyms. The BI Portal browser interface, or the Windows client, Developer Workbench.
- DataMigrator, a companion data warehouse building product, can use synonyms created by the above mentioned tools to represent source data to be replicated, or can also be used to create synonyms from scratch representing either source or target tables to populate.
- Independent software vendors (ISVs) might provide predefined metadata (and also reports) for their application package that you can import into DB2 Web Query and begin using immediately.
- An add-on feature of DB2 Web Query (Standard Edition) is a JDE (Oracle JD Edwards World and Enterprise ONE) adapter that provides pre-built synonyms and interfaces to the JDE data out of the box.

To create synonyms you must be a member of the DBA group for each top level folder of interest (done through security center). To use Developer Workbench for creating/enhancing synonyms, you must also be in the global DEV-Workbench group (see Chapter 6, “IBM DB2 Web Query for i Security Center” on page 127.

4.2.1 DB2 Web Query for i Metadata interfaces

There are three interfaces, or methods, that you can use to create synonyms for DB2 Web Query for i:

1. Developer Workbench.
2. The BI Portal browser interface.
3. Run the **CRTWQSYN** CL command.

The CL command is primarily for ISVs that want to deploy synonyms they have created over their data to their community of clients. It can be used to create synonyms over many files at once, but it provides little to no customization of the synonyms and simply projects the file’s definitions. For instance, field names are shown just as they are in the database.

The browser interface is very useful for syonym creation, simple editing, and lookup. For the most features, Developer Workbench is the most robust user interface for working with synonyms.

For an example showing how to create a synonym by using Developer Workbench, see section 4.4, “Example: Creating metadata with Developer Workbench” on page 61.

An alternative to starting from scratch in building metadata and reports, you may find the “metadata WIZARD” function to be of value in getting a quick start to building synonyms over your data. Through the WIZARDS, you can be walked through steps to create synonyms and 30-40 sample reports within minutes. Often, these sample reports can serve as a model for your own custom reports. To see the WIZARDS in action, you may want to watch this video demonstration!

4.3 Editing and enhancing metadata

The Developer Workbench and the web browser metadata interface both have a Synonym Editor that you can use to further refine and enhance them. Using the Synonym Editor, you can:

- Standardize column formatting (add commas and decimal positions, display values with % or currency symbol, and so on) and column titles.
- Define tablejoins.
- Convert and standardize date fields, and extract date components (Year, Quarter, Month, Day, and so on).
- Build virtual columns that standardize and centralize business logic (IF-THEN-ELSE business logic), or add options for filtering or grouping your data (create a REGION virtual column).
- Create filters (Boolean fields) for common or complex data filtering requests.
- Hide columns that report developers do not need.
- Calculate begin-end period information based on the current date (YTD, Prior YTD, last four rolling quarters, and so on).

The features and functions that are available in the Synonym Editor are the keys to fully leveraging the major benefits of abstracting data sources by using metadata: simplify, standardize, and enhance.

4.4 Example: Creating metadata with Developer Workbench

This example uses the Synonym Editor within Developer Workbench.

Complete the following steps:

1. Open Developer Workbench, right-click the system on which you want to work, and sign in with your user ID and password (Figure 4-1 and Figure 4-2).

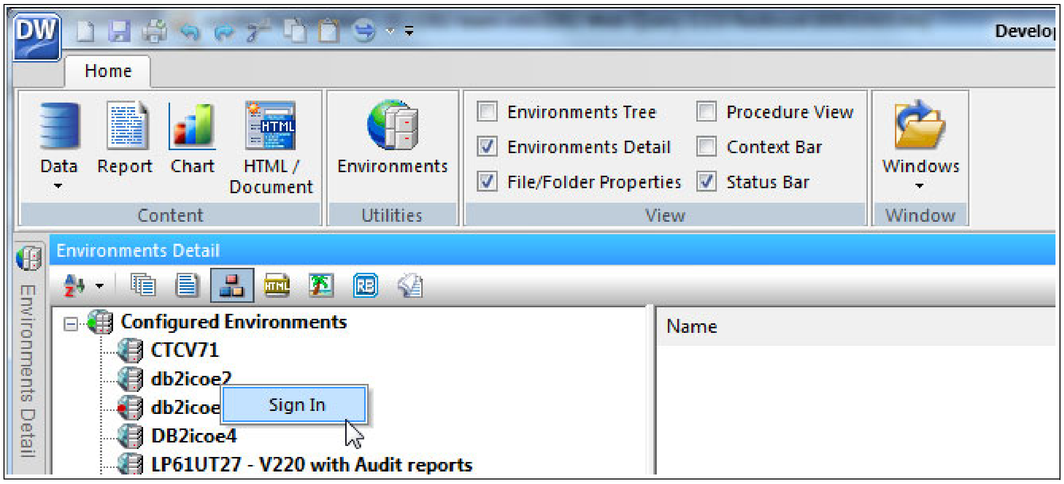
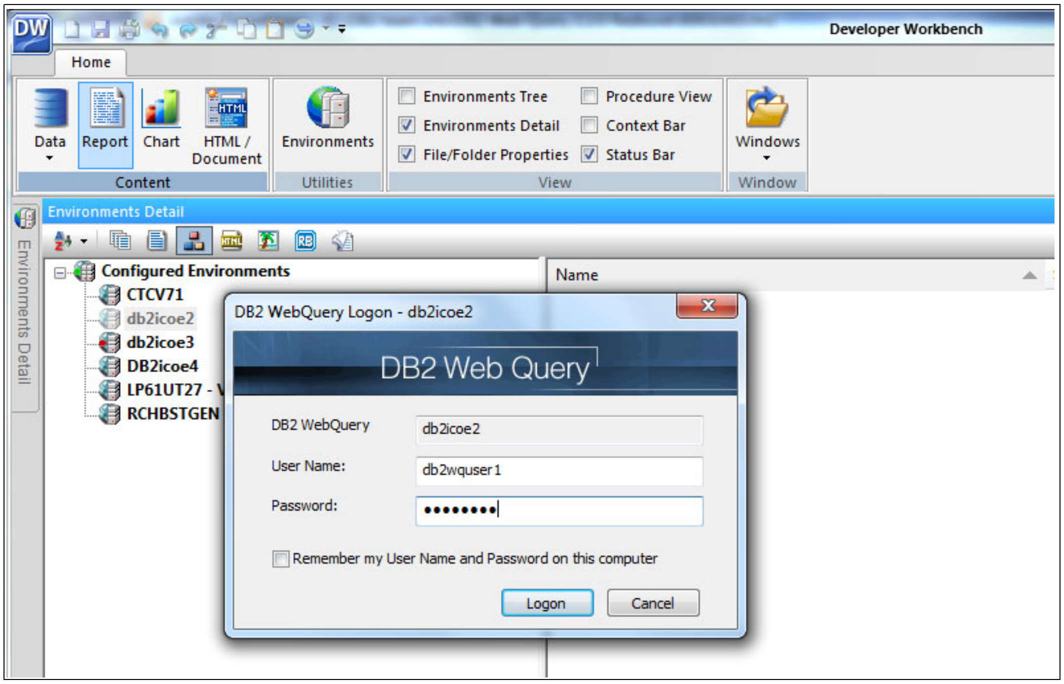


Figure 4-1 Select a system for sign-in



DB2 WebQuery: db2icoe2

User Name: db2wquser1

Password: ••••••••

☐ Remember my User Name and Password on this computer

Figure 4-2 Sign in to DB2 Web Query for i from Developer Workbench

2. The Developer Workbench window opens (Figure 4-3).

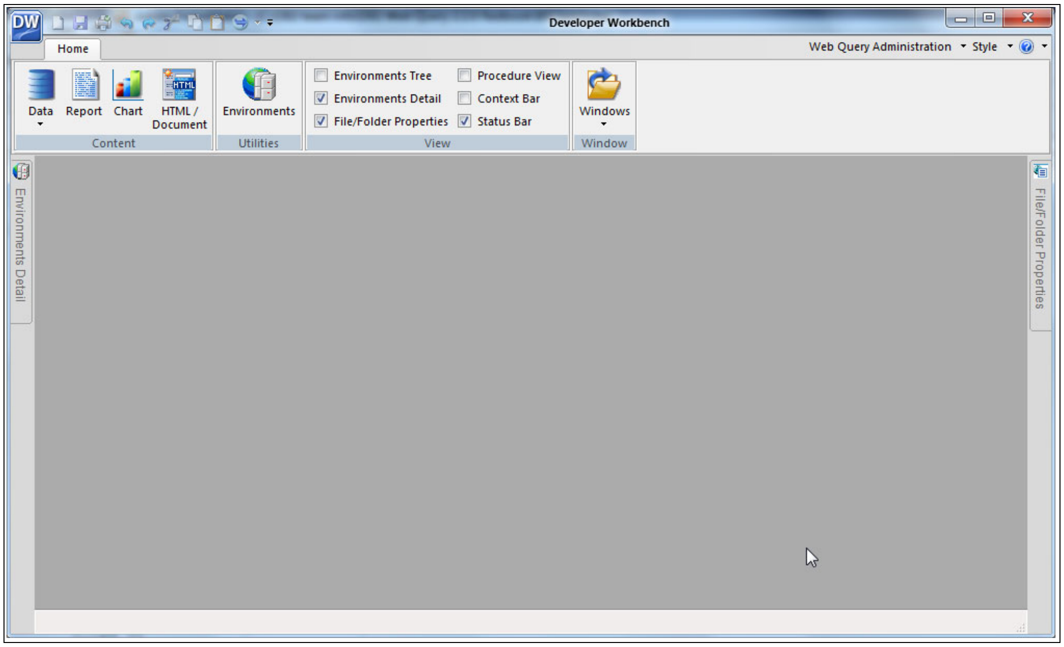


Figure 4-3 Default view of the Developer Workbench window after signing in

3. Click the **Environment Detail** tab and you see the configured systems. The system that you are signed in to is expanded to show the DB2 Web Query for i and Data Servers folders (Figure 4-4).

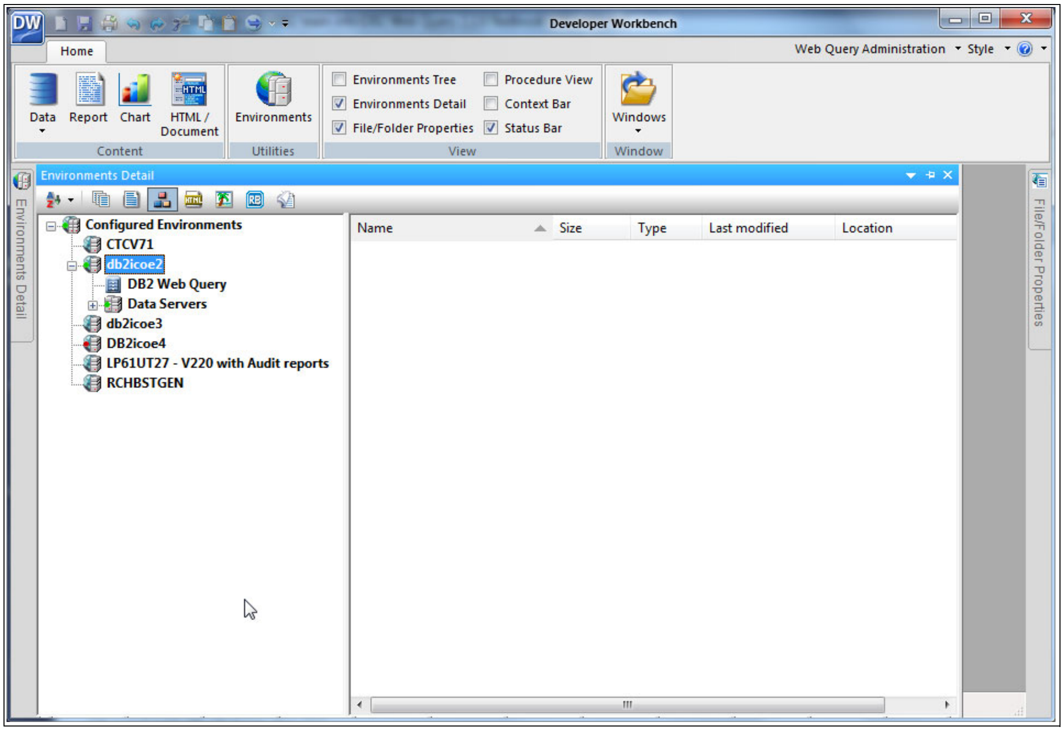


Figure 4-4 Environment Detail tab

4. Double-click to expand the **Data Servers** → **EDASERVE** → **Applications** folder structure. The application folders that you are authorized to access as a database administrator are shown in Figure 4-5.

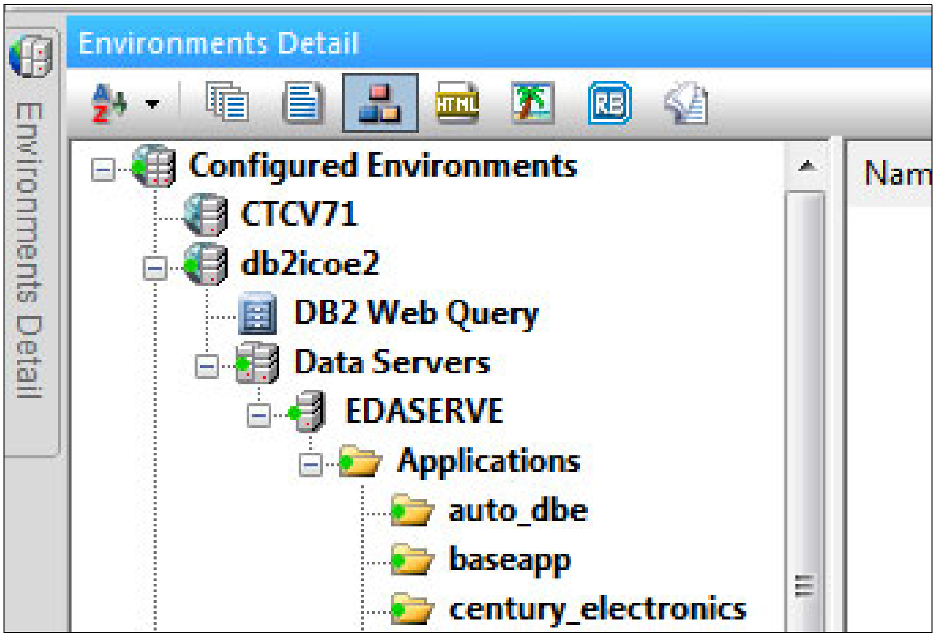


Figure 4-5 View of the authorized application folders

5. Right-click the folder where you want to create the synonym and select **New** → **Synonym**, as shown in Figure 4-6.

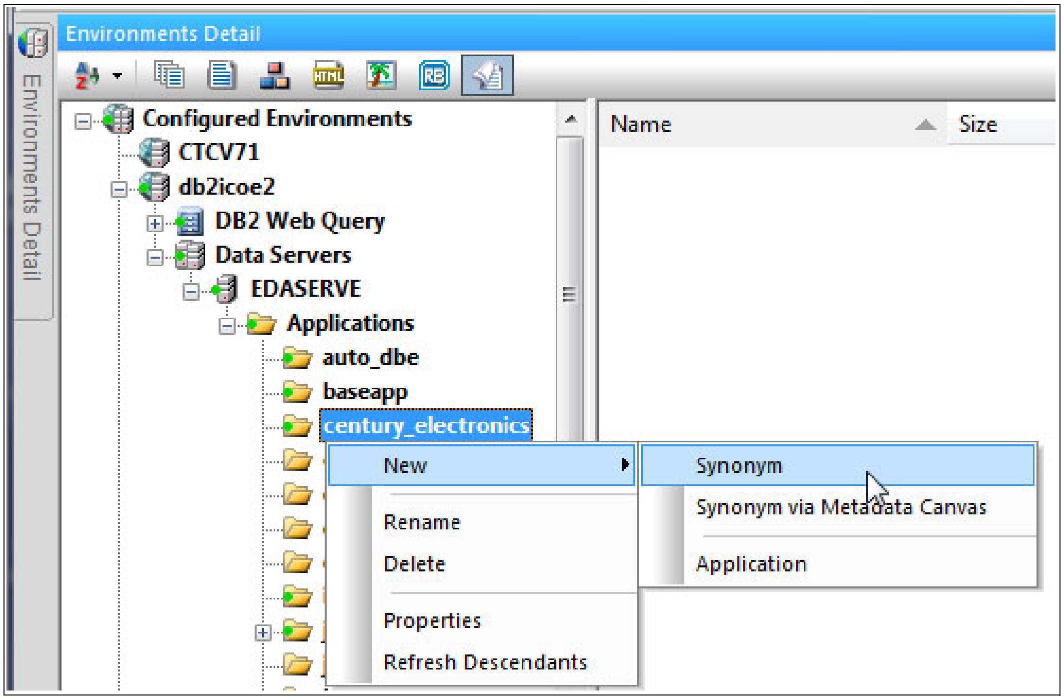


Figure 4-6 Create a synonym

6. The Select Server Node window opens. Select your target application folder and click **Select** (Figure 4-7).

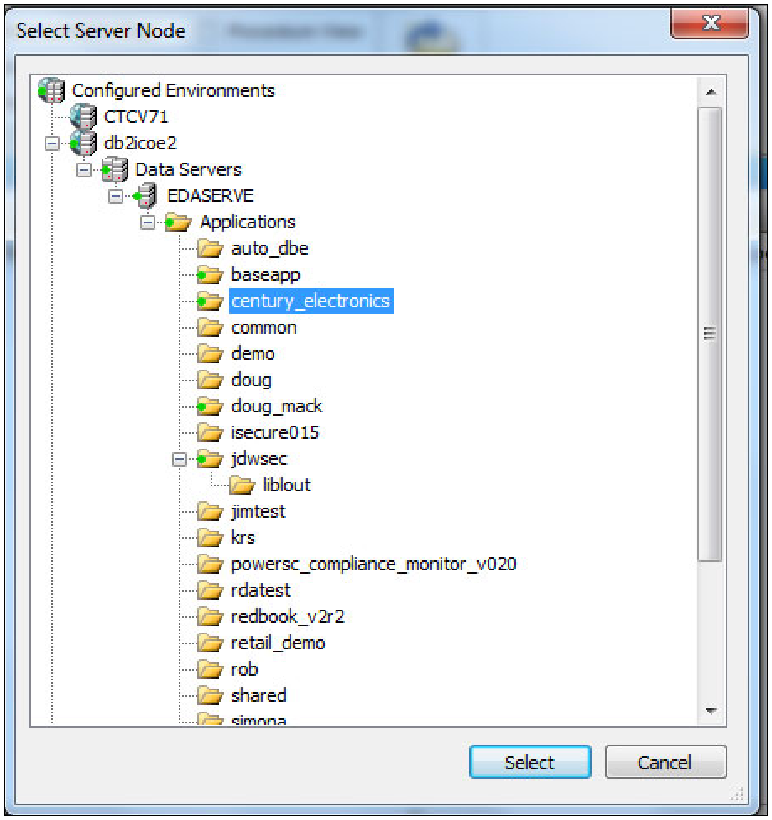


Figure 4-7 Select Server Node window

7. The Select Adapter window opens. Select ***LOCAL** and click **OK** (Figure 4-8). Note the other data sources available to you, including Query/400 definitions, and Excel, or if you expand on AVAILABLE you might see MS SQLServer, Postgres, MySQL and a generic JDBC driver.

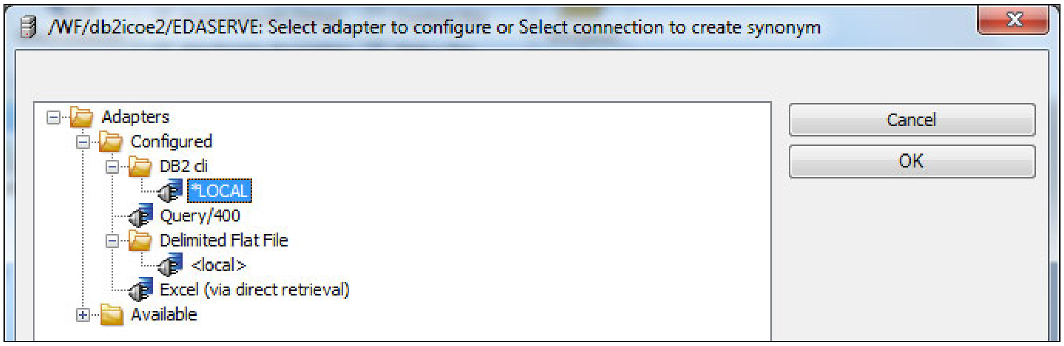
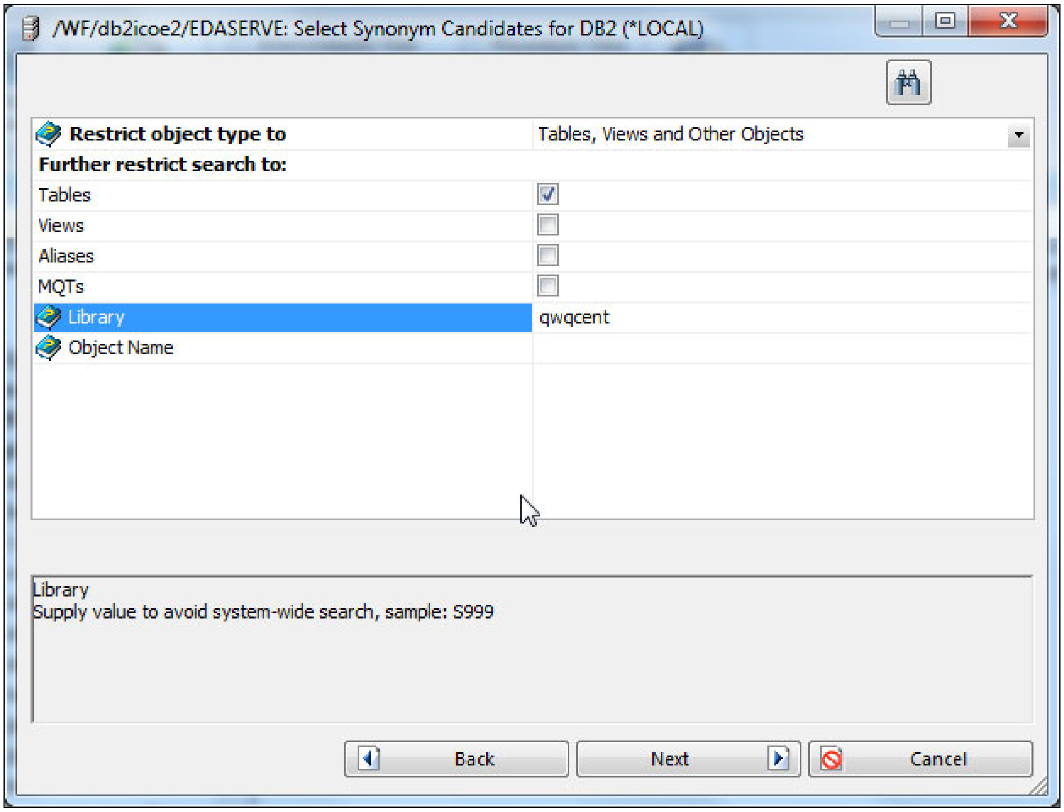

/WF/db2icoe2/EDASERVE: Select adapter to configure or Select connection to create synonym
Adapters
Configured
DB2 cli
*LOCAL
Query/400
Delimited Flat File
<local>
Excel (via direct retrieval)
Available
Cancel
OK

Figure 4-8 Select the adapter to create a synonym

8. In the Select Synonym Candidates for DB2 window, select the **Tables** check box. Place your cursor into the Library field and type in the name of the library that you want to search (QWQCENT in this example). Click **Next**, as shown in Figure 4-9.



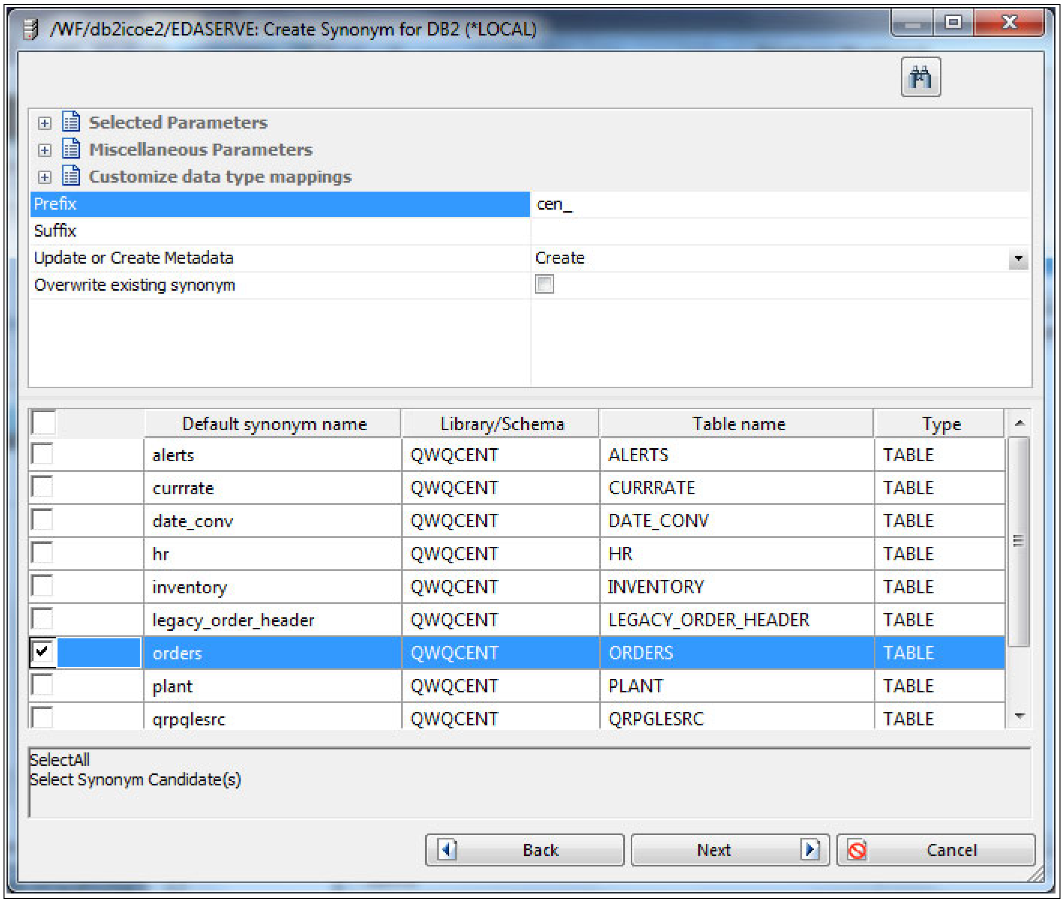
/WF/db2icoe2/EDASERVE: Select Synonym Candidates for DB2 (*LOCAL)

Restrict object type to	Tables, Views and Other Objects
Further restrict search to:	
Tables	☑
Views	□
Aliases	□
MQTs	□
Library	qwqcent
Object Name	

Library
Supply value to avoid system-wide search, sample: S999

Figure 4-9 Select Synonym Candidates for DB2

9. A window opens where you can choose the tables to use to create synonyms. The same considerations that are described in 4.4, “Example: Creating metadata with Developer Workbench” on page 61 apply here. In this example, we use the prefix `cen_` and select table orders, as shown in Figure 4-10. When you are done, click **Next**.



/WF/db2icoe2/EDASERVE: Create Synonym for DB2 (*LOCAL)

⊞ Selected Parameters
⊞ Miscellaneous Parameters
⊞ Customize data type mappings

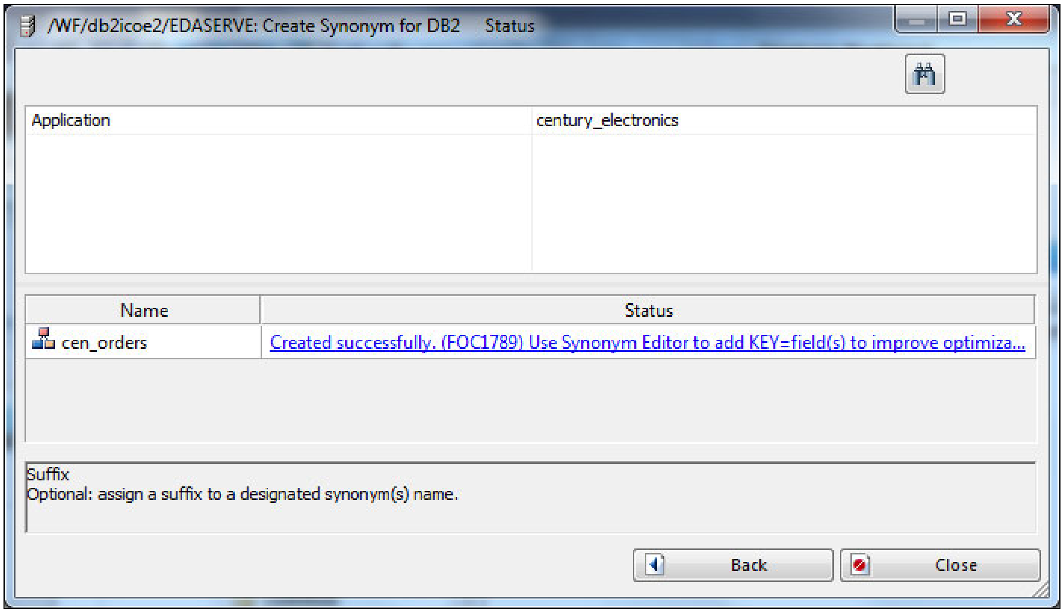
Prefix	cen_
Suffix	
Update or Create Metadata	Create
Overwrite existing synonym	□

□	Default synonym name	Library/Schema	Table name	Type
□	alerts	QWQCENT	ALERTS	TABLE
□	currrate	QWQCENT	CURRRATE	TABLE
□	date_conv	QWQCENT	DATE_CONV	TABLE
□	hr	QWQCENT	HR	TABLE
□	inventory	QWQCENT	INVENTORY	TABLE
□	legacy_order_header	QWQCENT	LEGACY_ORDER_HEADER	TABLE
☑	orders	QWQCENT	ORDERS	TABLE
□	plant	QWQCENT	PLANT	TABLE
□	qrpglesrc	QWQCENT	QRPGLESRC	TABLE

SelectAll
Select Synonym Candidate(s)

*Figure 4-10 Create Synonym for DB2 (*LOCAL)*

10. You are presented with a status message confirming the synonym creation (Figure 4-11). Click **Close**.



/WF/db2icoe2/EDASERVE: Create Synonym for DB2   Status

Application	century_electronics

Name	Status
cen_orders	Created successfully. (FOC1789) Use Synonym Editor to add KEY=field(s) to improve optimiza...

Suffix
Optional: assign a suffix to a designated synonym(s) name.

Figure 4-11   Create Synonym Status window

From here you could enhance the synonyms that you created with the synonym editor. You could:

- Build default formats of fields (for instance, anytime REVENUE is used it should include a “$” sign).
- Define tablejoins.
- Enhance non date data type fields from the legacy files into date attributes, such as DAY_OF_WEEK or TODAY-30.
- Build virtual columns that standardize and centralize business logic (IF-THEN-ELSE business logic), or add options for filtering or grouping your data (create a REGION virtual column).
- Create filters (Boolean fields) for common or complex data filtering requests.
- Hide columns that report developers do not need.

Calculate begin-end period information based on the current date (YTD, Prior YTD, last four rolling quarters, and so on).

5

Date and time functions

Date and time components are critical to querying and reporting. DB2 Web Query for i provides various ways to provide these key elements in your reporting environment. This chapter contains many of the details that are needed for including date and time elements in your reports and charts.

5.1 The importance of dates in reporting

Dates and times are vital business dimensions that are required in virtually all business reports. After all, how often do you create a report against data in your database without filtering, sorting, or aggregating the data against a date-related or time-related field? Ask any report developer what kinds of report requests they receive from their users, and you receive responses that include date-based attributes. Dates, time stamps, date-based attributes (year, quarter, month, season, and so on), and time stamp-based attributes (hour, minute, second, and so on) are critical because all reports have some aspect of a date dimension in their criteria. Here are some examples:

- A report showing the Current Year's product revenue and margin, by Quarter.
- A bar chart comparing Current Year Quarter to Date to Prior Year Quarter to Date.
- A *running total* chart with 13 monthly columns. Each column contains a listing of several key accounting measures, and the months that are shown in the report are based on the date that the report was run.
- A line chart showing, by customer and distribution center by week, the order shipment counts grouped by the ones that occurred on time versus those that were late (perhaps not obvious, but *on time* is typically defined as a calculation of the actual ship date being less than or equal to the requested or scheduled ship date).

At this point, you might be thinking: "I understand the importance and value of using date and date-based attributes in my reports, and agree that I need them. Unfortunately, most, if not all, of the database columns that I use for my reports are not true DATE data type columns. Can DB2 Web Query handle my legacy dates in Packed 8 Decimal, or 6 character without a century indicator, or Julien dates in numeric datatype fields?"

When it comes to how date values are being stored in the database files, you are not alone. A great many IBM i shops still use legacy date data types. Legacy dates are fields that are defined as numeric or alphanumeric fields and contain numbers or character strings that represent a date. An example of this is a field that is defined as zoned decimal (8,0), which contains the value 06152012 to represent the date June15, 2012. Business applications were written to treat these non-date columns as dates inside their program logic. However, any other application, this becomes very problematic and cumbersome to convert, parse out parts of the date you want, and apply functional logic to really get the date attribute that you want. Wouldn't it be easier if a report author could just drag a field onto their report called DAY_OF_WEEK or build a filter over FISCAL_QUARTER or even add TODAY-30 as an additional comparison column? Absolutely, and this is a great example of where metadata can be used to create those date attributes based on legacy dates and keep that information in one place, minimizing the complicated processes a report author would otherwise have to do.

Section 5.2, "Handling legacy dates" on page 71, describes different methods that you can use to "convert" legacy date formats into fields that DB2 Web Query recognizes as true date fields.

After it recognizes a field as being a date, DB2 Web Query for i can then provide additional reporting capabilities:

- Date decomposition can be used to break the date into separate fields that represent the year, quarter, month, and day. These new values can then be used in your reports for filtering, sorting, and grouping. For example, you could use the year component as a filter (filter on current year) and then show sums for a set of values by quarter (add quarter to the report's Across section).
- The date fields can be used for advanced date and time manipulation, calculations, and analysis. For example, you can add or subtract one year from a date column, or calculate the number of days between two dates.
- Report selection parameters that can be specified by invoking calendar widgets (date picker icons) for a more user-friendly experience.

All of these capabilities enable the report developer to deliver a report that is easy to use and to provide the report content and formatting that are required.

5.2 Handling legacy dates

This section describes four different methods for deriving a true date value from an existing legacy date value.

The first method involves modernization of the database files themselves, and is the most disruptive method in terms of the changes that are required to the existing database. The next two methods deal with conversion of the legacy dates into true dates by using programmatic means and involve using either the DB2 Web Query for i functions or DB2 database capabilities. The fourth method describes a technique that uses a DB2 database date dimension table that, when joined to your existing tables and their legacy dates, produces a true date or any of its date attributes.

Regardless of the method that is chosen, the result is the same: The legacy date is replaced with a field that DB2 Web Query can now process as a true date value. However, for most of you method 4, the Date Dimension table, will be the most effective choice.

5.2.1 Modernizing your database

If you are designing a new application and the data to support it, the DATE data type is always used whenever a field is used to hold date information. The database can then ensure that only proper dates are being entered and saved, and it can use the date-based features that are described earlier (for example, finding the difference in days between two dates). If the data modeling includes a data warehouse or data mart for reporting purposes, a temporal dimension also is created, that is, a dimension table that contains useful time-based attributes such as date, year, quarter, fiscal quarter, and month. The time dimension table, which is combined with the true date columns in the operational data, means that you can construct queries that quickly constrain your results to a given set of rows meeting your date and time requirements, and then perform additional processing such as eliminating additional records based on other criteria, providing totals by selected field groupings, and sorting the summed results.

Because you are dealing with existing structures containing legacy dates, you cannot design your database from the ground up. However, you can modify the existing database tables and add a true date column for each of your legacy date columns. You then create programs that convert legacy dates to true dates to populate the new columns. You also must add trigger programs to ensure that changes that are made to the legacy dates are reflected in their true date equivalents. With this in place, you now have date columns that are used as true dates by DB2 Web Query.

But again, this method involves significant changes to the database and its applications. Thus, unless a company is already undergoing a database modernization initiative, this is not the method that most companies select unless they convert those legacy dates to true date feilds are part of the transformation process in building data warehouses or data marts.

5.2.2 Using DB2 Web Query for i functions to convert to smart dates

This second method involves using the DB2 Web Query built-in functions (BIFs) to convert the legacy date field into its true date equivalent. The resulting date field is referred to as a *smart date*, meaning that it is a field that DB2 Web Query understands should be treated as a true date field. A smart date can be used just like a true date, in date calculations, decomposed into its date components (year, quarter, and so on) and formatted in a myriad of ways.

Smart dates are dynamically converted every time a report that uses the smart dates is run; they are not stored or saved between runs. This conversion process affects the overall report run time.

DATECVT is the most common BIF that is used when converting legacy dates to smart dates. This function converts the field value of any standard or legacy date format into a new date, in either the needed standard or legacy format.

Here is an example of using DATECVT:

```
DATECVT( ORDDAT, 'P8MDYY', 'MDYY' )
```

Where:

- `ORDDAT` is the name of the legacy field being converted.
- `‘P8MDYY’` is the structure of the ORDDAT legacy field (packed decimal (8,0) with MMDDYYYY structure).
- `‘MDYY’` is the output structure of the converted smart date (Smart Date with MMDDYYYY structure).

More examples of using DATECVT to create smart dates are shown in the Table 5-3 on page 104.

In the following exercise, you create a report from a file that uses packed decimal (8,0) date fields in the MDYY format. For example, the value 04102008 is used to represent the date April 10, 2008. You use the DB2 Web Query BIFs to convert these fields to virtual fields that are defined as a true date. This true date field is used as the basis for other virtual columns in other formats, such as year and day of the week.

> **Attention:** You also can create these virtual fields in the synonym (rather than in the report). In fact, this is the preferred approach, but for the purpose of this exercise, the conversions are performed in the report definition.

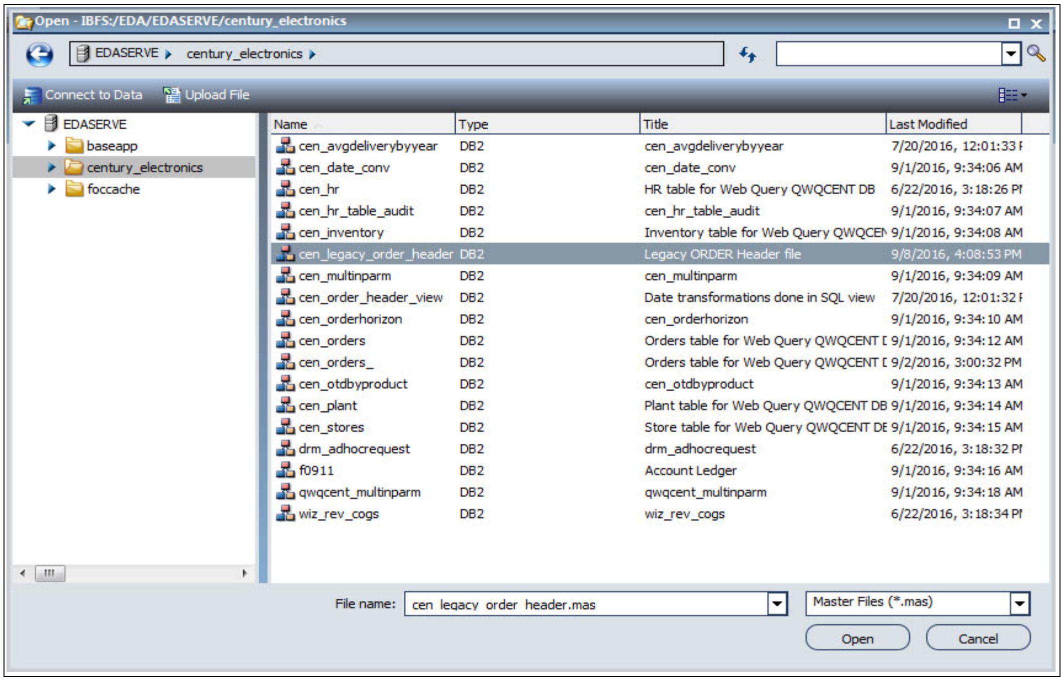
The report uses these virtual fields in a two-dimensional format:

- Sorted/grouped vertically by the year of the order
- Sorted/grouped across by the name of the day of the week

The measure that is shown for each of these groupings is the aggregated order amount.

To perform date conversions by using the DB2 Web Query BIFs, complete the following steps:

1. Open DB2 Web Query in a browser session and log in.
2. Create a synonym over the LEGACY_ORDER_HEADER table. Give it a prefix of `cen_`.
3. Create a report in the Century Electronics folder.
4. From the list of displayed synonyms, select **CEN_LEGACY_ORDER_HEADER** as the data source, as shown in Figure 5-1.


Open - IBFS:/EDA/EDASERVE/century_electronics
EDASERVE ▸ century_electronics ▸
Connect to Data
Upload File
EDASERVE
baseapp
century_electronics
foccache

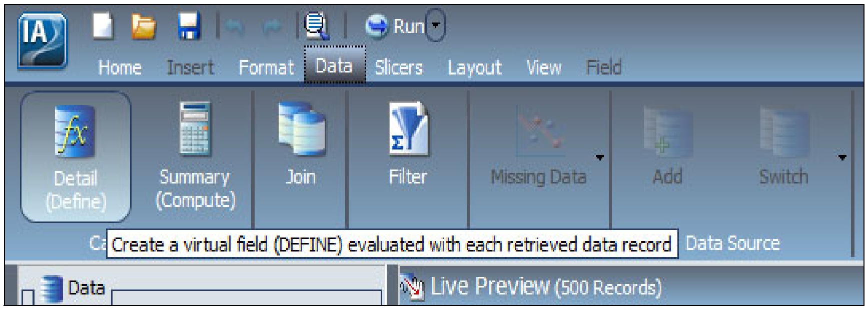
Name	Type	Title	Last Modified
cen_avgdeliverybyyear	DB2	cen_avgdeliverybyyear	7/20/2016, 12:01:33 F
cen_date_conv	DB2	cen_date_conv	9/1/2016, 9:34:06 AM
cen_hr	DB2	HR table for Web Query QWQCENT DB	6/22/2016, 3:18:26 PI
cen_hr_table_audit	DB2	cen_hr_table_audit	9/1/2016, 9:34:07 AM
cen_inventory	DB2	Inventory table for Web Query QWQCEN	9/1/2016, 9:34:08 AM
cen_legacy_order_header	DB2	Legacy ORDER Header file	9/8/2016, 4:08:53 PM
cen_multinparm	DB2	cen_multinparm	9/1/2016, 9:34:09 AM
cen_order_header_view	DB2	Date transformations done in SQL view	7/20/2016, 12:01:32 F
cen_orderhorizon	DB2	cen_orderhorizon	9/1/2016, 9:34:10 AM
cen_orders	DB2	Orders table for Web Query QWQCENT [	9/1/2016, 9:34:12 AM
cen_orders_	DB2	Orders table for Web Query QWQCENT [	9/2/2016, 3:00:32 PM
cen_otdbyproduct	DB2	cen_otdbyproduct	9/1/2016, 9:34:13 AM
cen_plant	DB2	Plant table for Web Query QWQCENT DB	9/1/2016, 9:34:14 AM
cen_stores	DB2	Store table for Web Query QWQCENT DE	9/1/2016, 9:34:15 AM
drm_adhocrequest	DB2	drm_adhocrequest	6/22/2016, 3:18:32 PI
f0911	DB2	Account Ledger	9/1/2016, 9:34:16 AM
qwqcent_multinparm	DB2	qwqcent_multinparm	9/1/2016, 9:34:18 AM
wiz_rev_cogs	DB2	wiz_rev_cogs	6/22/2016, 3:18:34 PI

File name: cen_legacy_order_header.mas

Master Files (*.mas)

Figure 5-1 Select synonym

5. In InfoAssist, create a Define field by clicking the **Data** tab and selecting the **Detail (Define)** icon, as shown in Figure 5-2.



Create a virtual field (DEFINE) evaluated with each retrieved data record

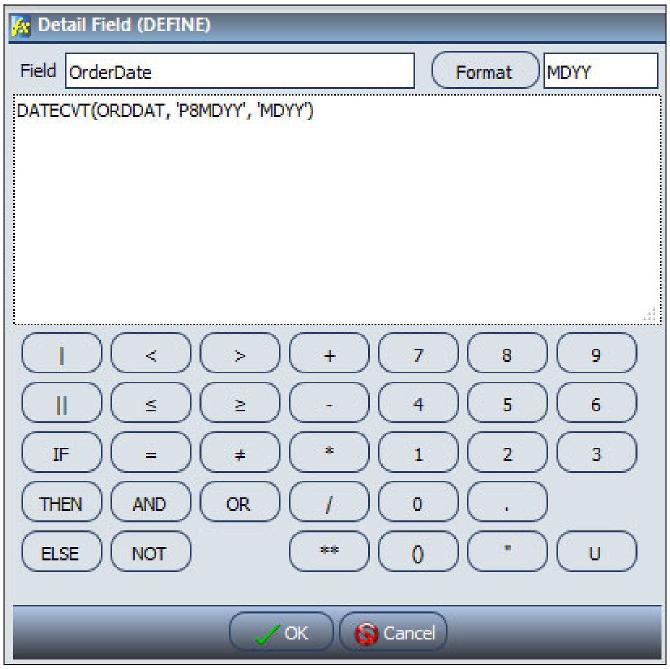
Figure 5-2 Create a Define field

To convert the packed decimal fields to dates, use the DATECVT function. This function converts the field value of any standard date format or legacy date format into a new date, in either the standard date format or the legacy date format. Here are the parameters for this function:

- The date is the input legacy field to be converted.
- The in_format is the format of the input legacy date (for example, P8MDYY, I8MDYY, I6YMD, and A8MDYY).
- The output_format is the output date format (for example, YYMD, YQ, M, DMY, and JUL).

6. In the Define Field (DEFINE) window, specify the following items, and then click **OK**:
   - Field: OrderDate (This is the new virtual column name.)
   - Format: MDYY (This is the output date format of the new virtual column.)
   - Expression: DATECVT(ORDDAT, 'P8MDYY', and 'MDYY')

   An example is provided in Figure 5-3.



Detail Field (DEFINE)

Field: OrderDate

Format: MDYY

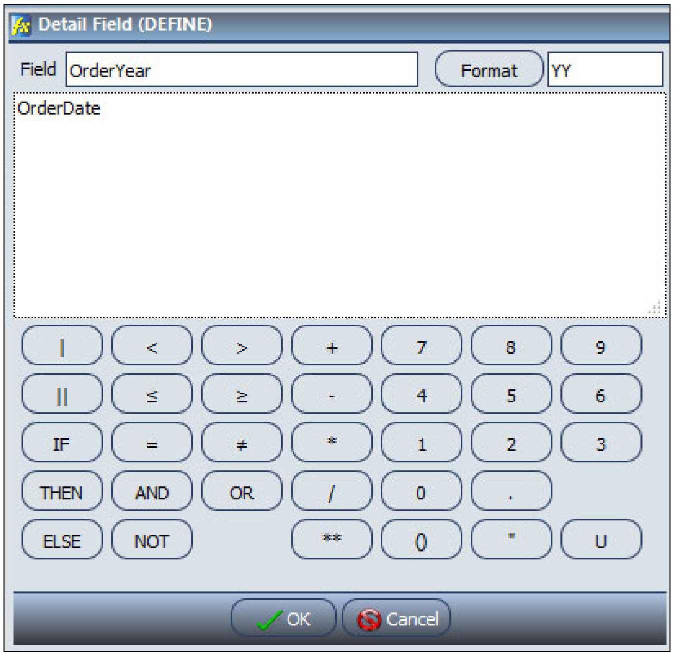
DATECVT(ORDDAT, 'P8MDYY', 'MDYY')

Figure 5-3 Define field for OrderDate

> **Note:** ORDDAT is a Packed Decimal (8,0) field. If you were diong this in a synonym for this field, you would notice that it is defined as P9 (rather than P8). This is normal. The extra digit is used to store the decimal point. For the purposes of date conversion, always ignore extra digits when specifying the value of the input format parameter. In this case, the value should be P8MDYY.

7. Create another Define field to display just the year of the OrderDate column. Specify the following attributes, then click **OK**:
   - Field: OrderYear
   - Format: YY
   - Expression: OrderDate

An example is provided in Figure 5-4.



Detail Field (DEFINE)

Field OrderYear

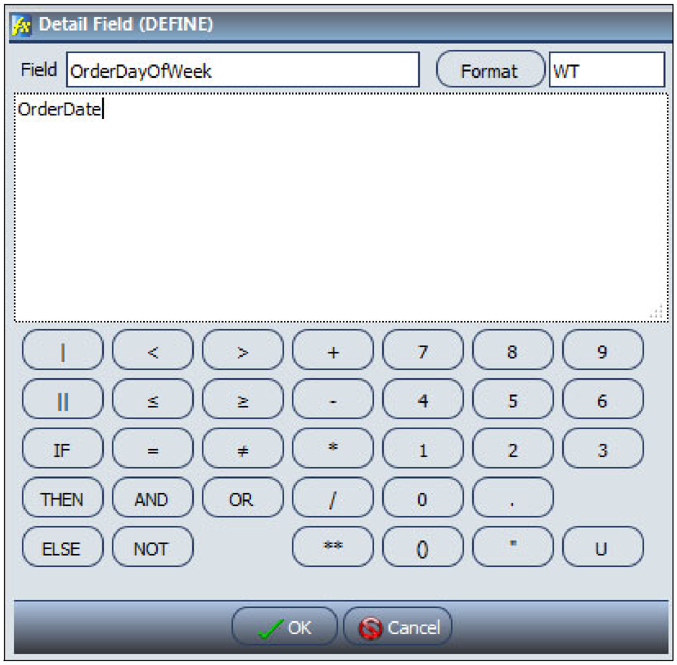
Format YY

OrderDate

Figure 5-4 Define field for OrderYear

8. Create another Define field to display the day of the week (MON, TUE, and so on) of the OrderDate column. Specify the following attributes, then click **OK**:
   - Field: OrderDayOfWeek
   - Format: WT
   - Expression: OrderDate

   An example is provided in Figure 5-5.



Detail Field (DEFINE)

Field OrderDayOfWeek

Format WT

OrderDate

Figure 5-5 Define field for OrderDayOfWeek

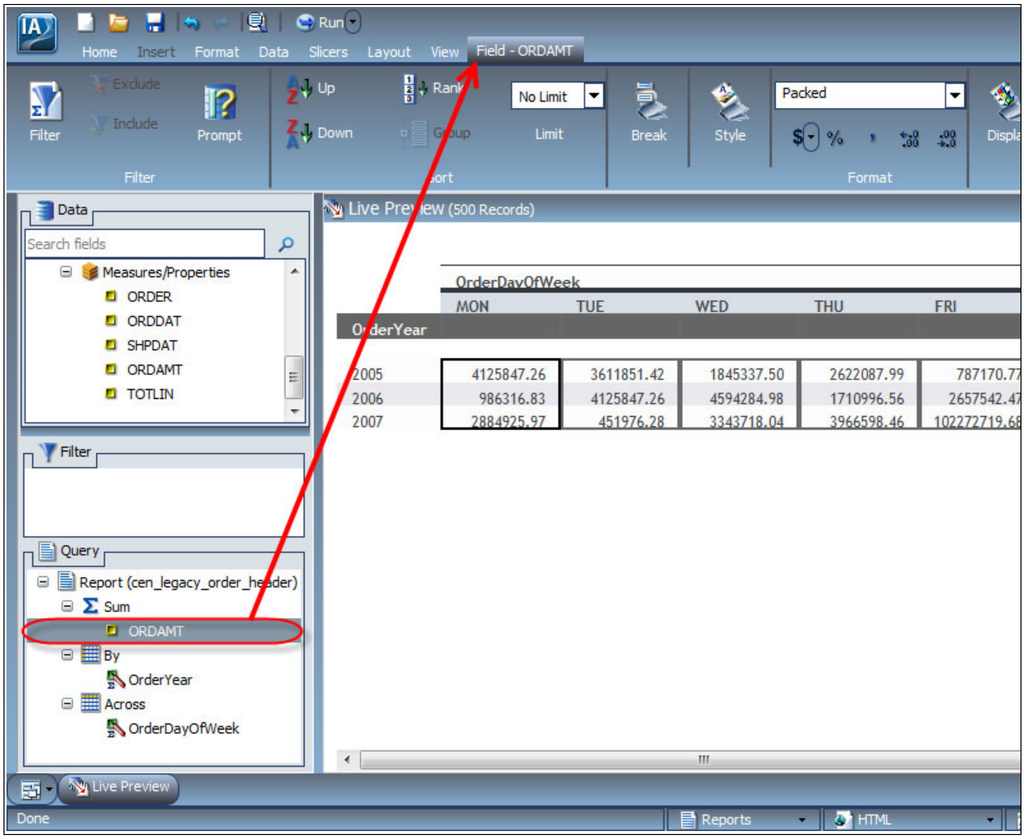
> **Attention:** A full list of date display formats can be found in 5.7.1, “Date format display options” on page 124.

9. Finish the report by completing the following steps:

   a. Drag the new OrderYear field into the Sort-By pane.

   b. Drag the OrderDayOfWeek field into the Sort across pane.

   c. Drag the ORDAMT field into the Sum pane. After dragging, click the Sum pane’s ORDAMT field, which opens the Field-ORDAMT options for this field, as shown in Figure 5-6.

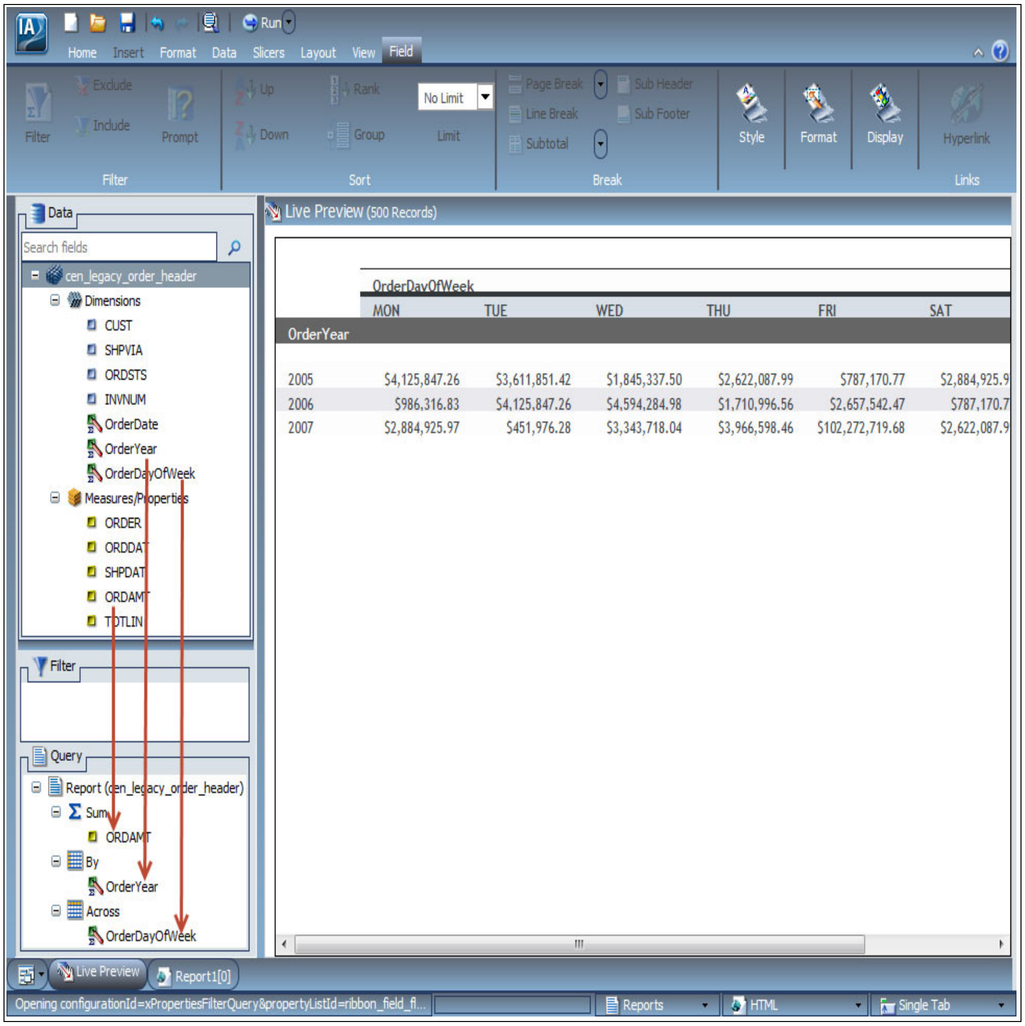


	OrderDayOfWeek				
	MON	TUE	WED	THU	FRI
OrderYear					
2005	4125847.26	3611851.42	1845337.50	2622087.99	787170.77
2006	986316.83	4125847.26	4594284.98	1710996.56	2657542.47
2007	2884925.97	451976.28	3343718.04	3966598.46	102272719.68

Figure 5-6   The Field-ORDAMT tree

   d. Edit the format of the ORDAMT field. Specify comma inclusion and floating currency.

e. When finished, the report definition should look like the example that is provided in Figure 5-7 or Figure 5-8 on page 78.

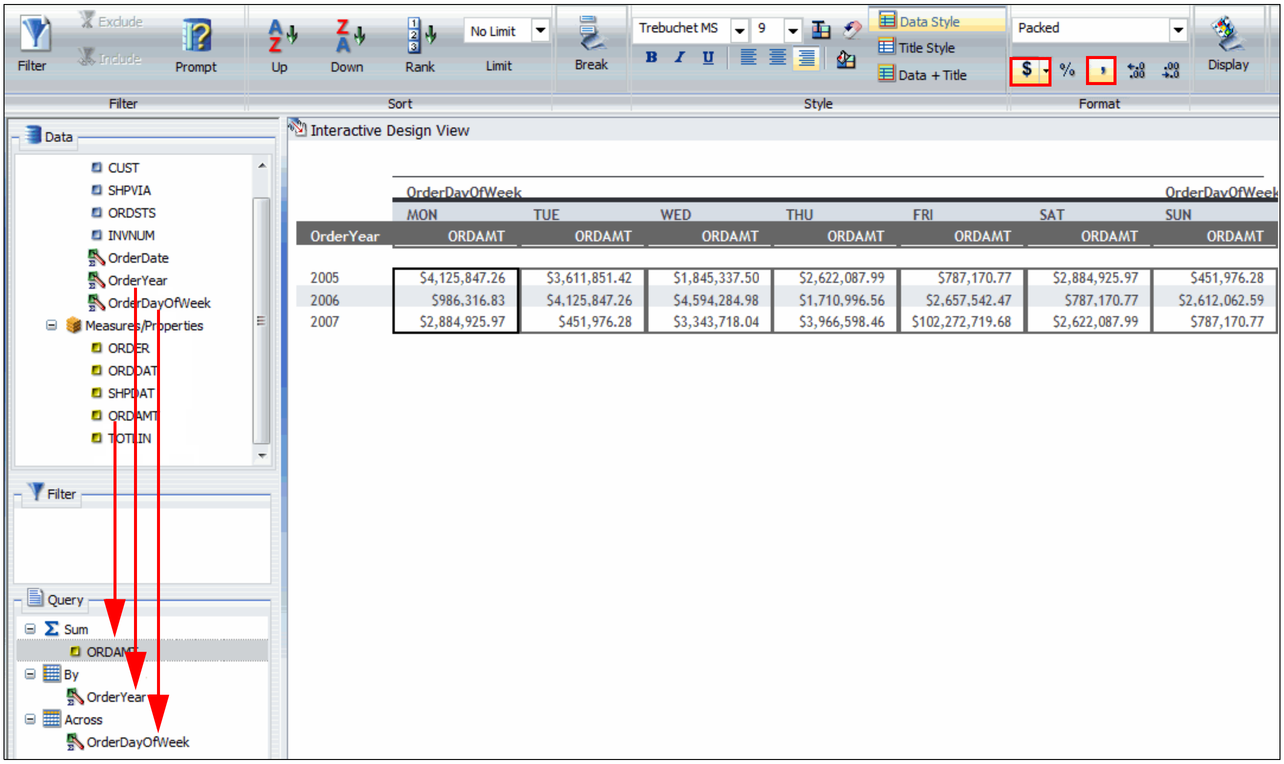


Live Preview (500 Records)

	OrderDayOfWeek					
	MON	TUE	WED	THU	FRI	SAT
OrderYear						
2005	$4,125,847.26	$3,611,851.42	$1,845,337.50	$2,622,087.99	$787,170.77	$2,884,925.9
2006	$986,316.83	$4,125,847.26	$4,594,284.98	$1,710,996.56	$2,657,542.47	$787,170.7
2007	$2,884,925.97	$451,976.28	$3,343,718.04	$3,966,598.46	$102,272,719.68	$2,622,087.9

Opening configurationId=xPropertiesFilterQuery&propertyListId=ribbon_field_fl...

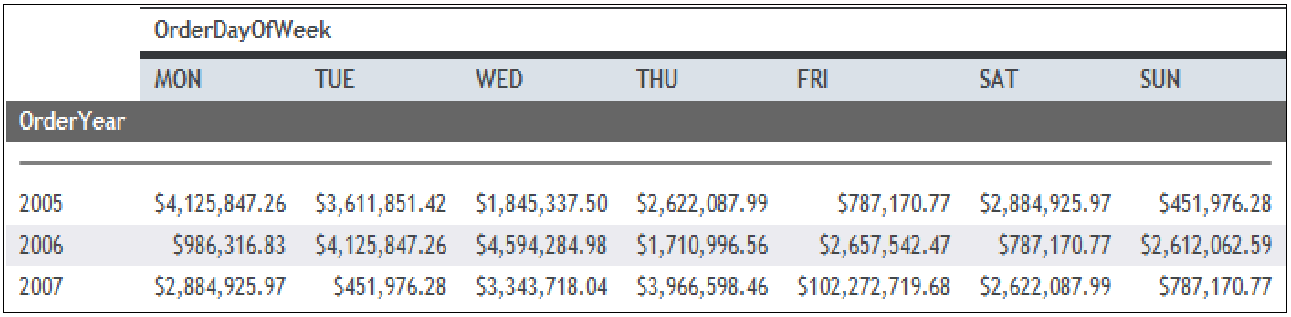
Figure 5-7 2.2.0 report that uses Define fields for date conversion



	OrderDayOfWeek						OrderDayOfWeek
	MON	TUE	WED	THU	FRI	SAT	SUN
OrderYear	ORDAMT	ORDAMT	ORDAMT	ORDAMT	ORDAMT	ORDAMT	ORDAMT
2005	$4,125,847.26	$3,611,851.42	$1,845,337.50	$2,622,087.99	$787,170.77	$2,884,925.97	$451,976.28
2006	$986,316.83	$4,125,847.26	$4,594,284.98	$1,710,996.56	$2,657,542.47	$787,170.77	$2,612,062.59
2007	$2,884,925.97	$451,976.28	$3,343,718.04	$3,966,598.46	$102,272,719.68	$2,622,087.99	$787,170.77

Figure 5-8 2.1.0 report that uses Define fields for date conversion

10. Run the report. It should look like the example in Figure 5-9.



	OrderDayOfWeek						
	MON	TUE	WED	THU	FRI	SAT	SUN
OrderYear							
2005	$4,125,847.26	$3,611,851.42	$1,845,337.50	$2,622,087.99	$787,170.77	$2,884,925.97	$451,976.28
2006	$986,316.83	$4,125,847.26	$4,594,284.98	$1,710,996.56	$2,657,542.47	$787,170.77	$2,612,062.59
2007	$2,884,925.97	$451,976.28	$3,343,718.04	$3,966,598.46	$102,272,719.68	$2,622,087.99	$787,170.77

Figure 5-9 Results of a report that uses Define fields for date conversion

11. Save your report as Date conversion by using Web Query functions.
12. Close InfoAssist.

5.2.3 Using SQL functions and views to convert dates

The third method for deriving a true date from a legacy date is to perform date conversions in the database by using the power of SQL functions built into DB2, and then reference these new true dates within SQL views. SQL functions such as **DATE**, **YEAR**, and **DAYOFWEEK** can be used in an SQL **SELECT** statement to efficiently perform the conversions. Depending on the legacy date data type and format, it might be necessary to use other functions, such as **DIGITS** and **SUBSTRING**. In some cases, this might result in a **SELECT** statement that is somewhat lengthy. But, you can hide this complexity from your report developers by creating an SQL view over this statement and then create your synonym over this SQL view.

An example SQL view that uses multiple SQL functions for date conversions (over the LEGACY_ORDER_HEADER table) is provided in the QWQCENT library. This view, which is named ORDER_HEADER_VIEW, is shown in Example 5-1.

Example 5-1   ORDER_HEADER_VIEW definition

```
CREATE VIEW ORDER_HEADER_VIEW AS
WITH t1 (
   order ,
   cust ,
   orderdate ,
   shipdate ,
   shpvia ,
   ordsts ,
   ordamt ,
   totlin ,
   invnum ) AS
   (SELECT order, cust,
      DATE(SUBSTRING(DIGITS(orddat),5,4) || '-' || SUBSTRING(DIGITS(orddat),1,2)
         || '-' || SUBSTRING(DIGITS(orddat),3,2)),
      DATE(SUBSTRING(DIGITS(shpdat),5,4) || '-' || SUBSTRING(DIGITS(shpdat),1,2)
         || '-' || SUBSTRING(DIGITS(shpdat),3,2)),
      shpvia, ordsts, ordamt, totlin, invnum
   FROM legacy_order_header )

SELECT order, cust, orderdate, YEAR(orderdate) AS orderyear,
   DAYOFWEEK_ISO(orderdate) AS orderdow,
   CASE
      WHEN DAYOFWEEK_ISO(orderdate) = 1 THEN 'MON'
      WHEN DAYOFWEEK_ISO(orderdate) = 2 THEN 'TUE'
      WHEN DAYOFWEEK_ISO(orderdate) = 3 THEN 'WED'
      WHEN DAYOFWEEK_ISO(orderdate) = 4 THEN 'THU'
      WHEN DAYOFWEEK_ISO(orderdate) = 5 THEN 'FRI'
      WHEN DAYOFWEEK_ISO(orderdate) = 6 THEN 'SAT'
      WHEN DAYOFWEEK_ISO(orderdate) = 7 THEN 'SUN'
      ELSE ''
   END AS orderdayname,
   shipdate, shpvia, ordsts, ordamt, totlin, invnum FROM t1
```

ORDER_HEADER_VIEW performs the following tasks:

- Converts legacy date fields ORDDAT and SHPDAT into true dates and returns them in the derived columns that are named ORDERDATE and SHIPDATE.
- Creates a second derived column that is named ORDERYEAR by referencing the derived column ORDERDATE in the SQL function YEAR. This column returns the year portion of the date.
- Creates a third derived column that is named ORDERDOW by referencing the derived column ORDERDATE in the SQL function DAYOFWEEK_ISO. This column returns the integer value of the day of the week.
- Creates a fourth derived column that is named ORDERDAYNAME by referencing the derived column ORDERDATE in the SQL function DAYOFWEEK_ISO and performing some logic in a `CASE` statement to return the name of the day of the week.

Notice the use of the **WITH** keyword to create SQL common table expressions. If you are not familiar with common table expressions, they can be thought of as temporary views that exist only during the execution of the query. By using them, you can define a result table with a table-identifier (T1 above) that can be specified as the table name in any **FROM** clause of the **SELECT** statement. Because the example uses derived columns that reference other derived columns (ORDERDATE), a common table expression is used. You can use this expression to define ORDERDATE in one place and reduce the query's complexity by minimizing the amount of text, especially when ORDERDATE is referenced multiple times in the query.

More examples of using SQL functions to convert legacy dates to true dates are shown in Table 5-3 on page 104.

In the following exercise, you create a report that uses the synonym that is built over QWQCENT/ORDER_HEADER_VIEW. This report is identical in appearance to the report built in 5.2.2, “Using DB2 Web Query for i functions to convert to smart dates” on page 72. The difference is that this time you use SQL functions to perform the conversions instead of using BIFs.

Complete the following steps:

1. Create a synonym over the ORDER_HEADER_VIEW view. Specify `cen_` for the metadata prefix.
2. Create a report in InfoAssist by using the cen_order_header_view synonym:
   a. Drag the ORDERYEAR field into the By field container in the Query pane.
   b. Drag the ORDERDOW field into the Across field container in the Query pane. Similar to the report in 5.2.2, “Using DB2 Web Query for i functions to convert to smart dates” on page 72, make it a hidden field so that the columns are sorted by the numeric value of the day of the week rather than alphabetically by the name of the day of the week.
   c. Drag the ORDERDAYNAME field into the Across field container in the Query pane. Make sure it is after ORDERDOW.
   d. Drag the ORDAMT field into the Sum field container in the Query pane and select it.
   e. Add comma inclusion and floating currency formatting to the ORDAMT field in the report.

When finished, the report definition should look like the example that is provided in Figure 5-10.

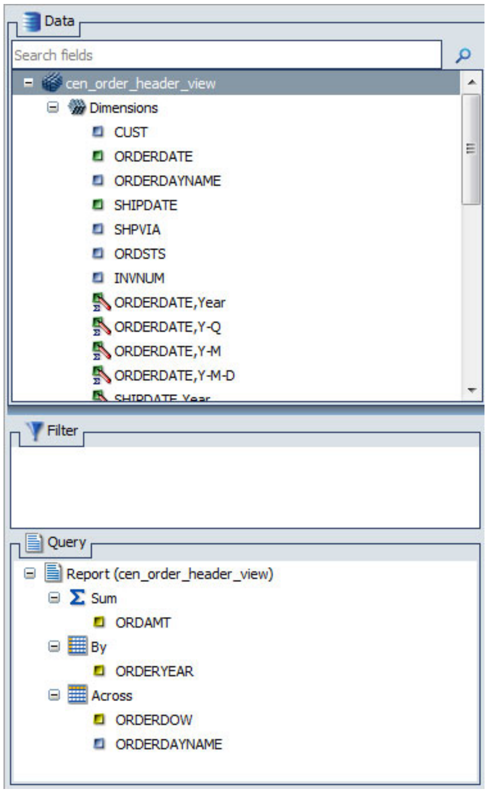
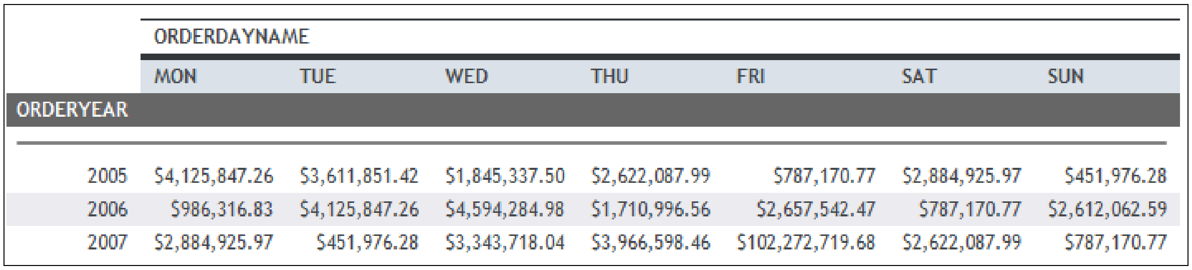


Figure 5-10 Report that uses the SQL view for date conversions

3. Run the report. It should look like the example that is provided in Figure 5-11.



	ORDERDAYNAME						
	MON	TUE	WED	THU	FRI	SAT	SUN
ORDERYEAR							
2005	$4,125,847.26	$3,611,851.42	$1,845,337.50	$2,622,087.99	$787,170.77	$2,884,925.97	$451,976.28
2006	$986,316.83	$4,125,847.26	$4,594,284.98	$1,710,996.56	$2,657,542.47	$787,170.77	$2,612,062.59
2007	$2,884,925.97	$451,976.28	$3,343,718.04	$3,966,598.46	$102,272,719.68	$2,622,087.99	$787,170.77

Figure 5-11 Results of report that uses the SQL view for date conversions

4. Save your report as a Date conversion by using SQL view.

5.2.4 Using a date dimension table

Also known as a date conversion table or a calendar table, a *date dimension* table is simply a DB2 for i table that contains one row for each individual day within some specified date range (for example, all days from January 1, 1900 through December 31, 2099). Each row contains a true date plus one or more legacy dates that are used by the business to represent the true date. The rest of the table consists of columns containing important date attributes for each date row, meaning that the attributes are useful for reporting in terms of constraining the data, providing new grouping and aggregation possibilities, to name a few. These date attribute columns might include the standard attributes of year, quarter, month and day, and also fiscal quarter, ISO week of the year, season of the year, and phase of the moon. Your reporting and business analytics requirements dictate how many legacy date value representations and date attributes you need to add to your date dimension table.

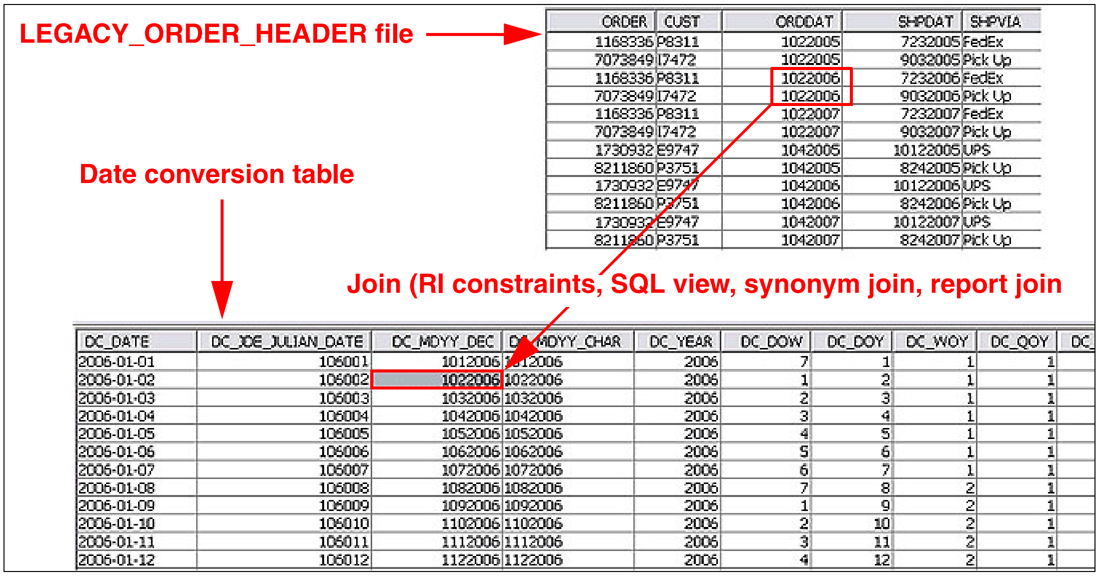
For example, a date dimension table might include the following columns:

- Julian legacy date, which is used when joining to files that use Julian legacy dates
- Packed decimal(8,0) legacy date, which is used when joining to files that use Packed decimal(8,0) legacy dates
- Character(8) legacy date, which is used when joining to files that use Character(8) legacy dates
- Date (a *true* DB2 date field)
- Fiscal year
- Fiscal quarter
- Day of the week (Monday, Tuesday, and so on)
- Month of the year (January, February, and so on)
- Season (spring, summer, autumn, and winter)
- Same day (of the week) last year
- Week ending date
- Week of the year
- Super Bowl Sunday flag (Y or N)
- Day before a holiday flag (Y or N)
- Day after a holiday flag (Y or N)
- Full moon flag (Y or N)

In order for this technique to work, the date dimension table must have a column that represents the date in the same format as the legacy file. You then define an inner join from the legacy file to the date dimension table by using the matching legacy date field types as the join columns. This task can be done with any of the following techniques:

- Implementing Referential Integrity (setting up Primary and Foreign Keys)
- Creating an SQL view with syntax to join your legacy files to the date dimension table
- Defining the join in the synonym
- Defining the join in each report/graph

When the report is run, DB2 Web Query uses the chosen join definition method to generate the SQL syntax that is necessary to join the legacy file and the date dimension table. For each row that is returned in the legacy file, the matching row (for that date) of the date dimension table is also returned, providing the report with all the various columns representing that particular date. The result is an efficient legacy date conversion implementation and a faster-running report. Figure 5-12 illustrates how the join to the date dimension table works.



ORDER	CUST	ORDDAT	SHPDAT	SHPVIA
1168336	P8311	1022005	7232005	FedEx
7073849	I7472	1022005	9032005	Pick Up
1168336	P8311	1022006	7232006	FedEx
7073849	I7472	1022006	9032006	Pick Up
1168336	P8311	1022007	7232007	FedEx
7073849	I7472	1022007	9032007	Pick Up
1730932	E9747	1042005	10122005	UPS
8211860	P3751	1042005	8242005	Pick Up
1730932	E9747	1042006	10122006	UPS
8211860	P3751	1042006	8242006	Pick Up
1730932	E9747	1042007	10122007	UPS
8211860	P3751	1042007	8242007	Pick Up

DC_DATE	DC_JDE_JULIAN_DATE	DC_MDYY_DEC	DC_MDYY_CHAR	DC_YEAR	DC_DOW	DC_DOY	DC_WOY	DC_QOY	DC_
2006-01-01	106001	1012006	1012006	2006	7	1	1	1	
2006-01-02	106002	1022006	1022006	2006	1	2	1	1	
2006-01-03	106003	1032006	1032006	2006	2	3	1	1	
2006-01-04	106004	1042006	1042006	2006	3	4	1	1	
2006-01-05	106005	1052006	1052006	2006	4	5	1	1	
2006-01-06	106006	1062006	1062006	2006	5	6	1	1	
2006-01-07	106007	1072006	1072006	2006	6	7	1	1	
2006-01-08	106008	1082006	1082006	2006	7	8	2	1	
2006-01-09	106009	1092006	1092006	2006	1	9	2	1	
2006-01-10	106010	1102006	1102006	2006	2	10	2	1	
2006-01-11	106011	1112006	1112006	2006	3	11	2	1	
2006-01-12	106012	1122006	1122006	2006	4	12	2	1	

Figure 5-12 Joining files to the date dimension table

With this technique, each date format (column) in the date dimension table is available to the development tools and therefore can be easily included in any report. The report developer can easily develop some interesting customized analysis. For example:

- What are the profit margins on days before and after holidays?
- How many bags of corn chips are sold on the day before the Super Bowl?
- What is the rate of product returns during the week after Christmas as compared to any other week of the year?
- Are more galoshes sold in the spring or the fall?
- How many hospital emergency room visits occur on days when there is a full moon?

The usefulness of a date dimension table goes beyond DB2 Web Query. It can be implemented in any application or tool that has access to DB2 for i. If you have RPG programs, you can join to this table (by using either embedded SQL or native Record Level Access operations such as CHAIN) to perform quick and easy date conversions/lookups. It can even be used in your Query/400 reports.

Implementing the date dimension table method is relatively simple and can be performed in the following four steps:

1. Create the date dimension table.
2. Populate and maintain the date dimension table.

3. Join the date dimension table to the table that contains your legacy date. To do this task, either create a synonym over the date dimension table and join it to another synonym that contains your legacy date fields, or you can create a DB2 SQL view that joins the dimension table to the table containing the legacy date and then create a corresponding synonym over this view.
4. Create reports by using the joined date attribute information.

Creating the date dimension table

A pre-populated date dimension table that is named DATE_CONV is included in the QWQCENT library. This library is shipped in a save file format when you install the DB2 Web Query for i product 5733WQX. The name of the save file is QWQCENT and it is in the library QWEBQRY. The QWQCENT library might be updated occasionally to provide more objects, columns, stored procedures, and other examples. So, you might want to occasionally check to see whether a new version has been provided and if so, restore it. At the time of writing, DATE_CONV contained the columns that are shown in Table 5-1.

Table 5-1 Date dimension table

Column name	Data length	Length	Description
DC_DATE	DATE	4	Date (date format)
DC_JDE_JULIAN_DATE	DECIMAL	6	JDE Julian Date (CYYDDD decimal)
DC_MDYY_DEC	DECIMAL	8	Date (MMDDYYYY packed decimal)
DC_MDYY_ZONED	NUMERIC	8	Date (MMDDYYYY zoned decimal)
DC_MDYY_CHAR	CHAR	8	Date (MMDDYYYY character)
DC_YYMD_DEC	DECIMAL	8	Date (YYYYMMDD packed decimal)
DC_YYMD_ZONED	NUMERIC	8	Date (YYYYMMDD zoned decimal)
DC_YYMD_CHAR	CHAR	8	Date (YYYYMMDD character)
DC_MDY_DEC	DECIMAL	6	Date (MMDDYY packed decimal)
DC_MDY_ZONED	NUMERIC	6	Date (MMDDYY zoned decimal)
DC_MDY_CHAR	CHAR	6	Date (MMDDYY character)
DC_YMD_DEC	DECIMAL	6	Date (YYMMDD packed decimal)
DC_YMD_ZONED	NUMERIC	6	Date (YYMMDD zoned decimal)
DC_YMD_CHAR	CHAR	6	Date (YYMMDD character)
DC_CC_CHAR	CHAR	2	Century (2 characters)
DC_YY_CHAR	CHAR	2	Year (2 characters)
DC_MM_CHAR	CHAR	2	Month (2 characters)
DC_DD_CHAR	CHAR	2	Day (2 characters)
DC_YEAR	INTEGER	4	Year (4 digits)
DC_DOW	INTEGER	4	Day of week (1 - 7)
DC_DOW_ISO	INTEGER	4	Day of week (1 - 7)
DC_DOY	INTEGER	4	Day of year (1 - 366)

DC_WOY	INTEGER	4	Week of year (1 - 52)
DC_WOY_ISO	INTEGER	4	Week of year (1 - 53)
DC_QOY	INTEGER	4	Quarter of year (1 - 4)
DC_CC	NUMERIC	2	Century (2 digits)
DC_YY	NUMERIC	2	Year (2 digits)
DC_MM	NUMERIC	2	Month (2 digits)
DC_DD	NUMERIC	2	Day (2 digits)
DC_CCYYMM	NUMERIC	6	Century, Year, Month CCYYMM (6 digits)
DC_DAY_NAME	CHAR	9	Day Name (Monday, and so on)
DC_QUARTER_NAME	CHAR	6	Quarter name (2008Q1)
DC_WEEKEND	CHAR	1	Weekend Flag (Y or N)
DC_HOLIDAY	CHAR	1	Holiday (Y or N)
DC_DAY_BEFORE_HOLIDAY	CHAR	1	Day Before Holiday (Y or N)
DC_DAY_AFTER_HOLIDAY	CHAR	1	Day AfterHoliday (Y or N)
DC_FULL_MOON	CHAR	1	Full Moon (Y or N)
DC_SEASON	CHAR	6	Season (Spring, Summer, Autumn, and Winter)
DC_FISCAL_YEAR	INTEGER	4	Fiscal year (4 digits)
DC_FISCAL_QUARTER	INTEGER	4	Fiscal quarter (1-4)
DC_MONTH_NAME	CHAR	9	Month name (January, and so on)
DC_MONTH_ABRV	CHAR	3	Month abbreviation (Jan, Feb, and so on)
DC_JULIAN	NUMERIC	7	Date in Julian format
DC_CYYMMDD	DECIMAL	7	CYYMMDD packed C = 0 for 1900 & C = 1 for 2000
DC_EXCEL_DATE	INTEGER	4	Date in Excel format
DC_WEEK_STARTING_DATE	DATE	4	Week starting date (the prior Saturday)
DC_WEEK_ENDING_DATE	DATE	4	Week ending date (the next Friday)
DC_SAME_DAY_LAST_YEAR	DATE	4	Same day last year
DC_CURRENT_DAY	CHAR	1	Current Day (Y/N)
DC_CURRENT_WEEK	CHAR	1	Current Week (Y/N)
DC_CURRENT_MONTH	CHAR	1	Current Month (Y/N)
DC_CURRENT_QUARTER	CHAR	1	Current Quarter (Y/N)
DC_CURRENT_YEAR	CHAR	1	Current Year (Y/N)
DC_CURRENT_YTD	CHAR	1	Current Year to Date (Y/N)
DC_CURRENT_DAY_LAST_YEAR	CHAR	1	Current Day Last Year (Y/N)

DC_CURRENT_WEEK_LAST_YEAR	CHAR	1	Current Week Last Year (Y/N)
DC_CURRENT_MONTH_LAST_YEAR	CHAR	1	Current Month Last Year (Y/N)
DC_CURRENT_QUARTER_LAST_YEAR	CHAR	1	Current Quarter Last Year (Y/N)
DC_CURRENT_YEAR_LAST_YEAR	CHAR	1	Current Year Last Year (Y/N)
DC_CURRENT_YTD_LAST_YEAR	CHAR	1	Current Year To Date Last Year (Y/N)
DC_PREVIOUS_DAY	CHAR	1	Previous Day (Y/N)
DC_PREVIOUS_WEEK	CHAR	1	Previous Week (Y/N)
DC_PREVIOUS_MONTH	CHAR	1	Previous Month (Y/N)
DC_PREVIOUS_QUARTER	CHAR	1	Previous Quarter (Y/N)
DC_PREVIOUS_YEAR	CHAR	1	Previous Year (Y/N)
DC_PREVIOUS_FISCAL_YEAR	CHAR	1	Previous fiscal year (Y/N)
DC_CURRENT_FISCAL_YEAR	CHAR	1	Current fiscal year (Y/N)
DC_PREVIOUS_FISCAL_YTD	CHAR	1	Previous fiscal year to date (Y/N)
DC_CURRENT_FISCAL_YTD	CHAR	1	Current fiscal year to date (Y/N)
DC_NTH_DAY_OF_WEEK_OF_MONTH	INTEGER	4	Nth Day of the Week of the month

Populating and maintaining the date dimension table

As mentioned previously, a pre-populated date dimension table is included in the QWQCENT library. Therefore, unless there are columns that you want to add or remove, you can simply use this version of the table. If you are interested in creating a customized version, the SQL stored procedure that is used to populate this table is also provided in the QWQCENT library. The name of this stored procedure is LOAD_DATE_CONVERSION_TABLE and it creates one row for each day on and between January 1, 1900 and December 31, 2030. If you do create your own Date Dimension tables, you should NOT store it in QWQCENT, otherwise there is the possibility it will get overwritten on an update to that library. You can see the SQL source code of this stored procedure by using the Generate SQL option from System i Navigator, as shown in Figure 5-13.

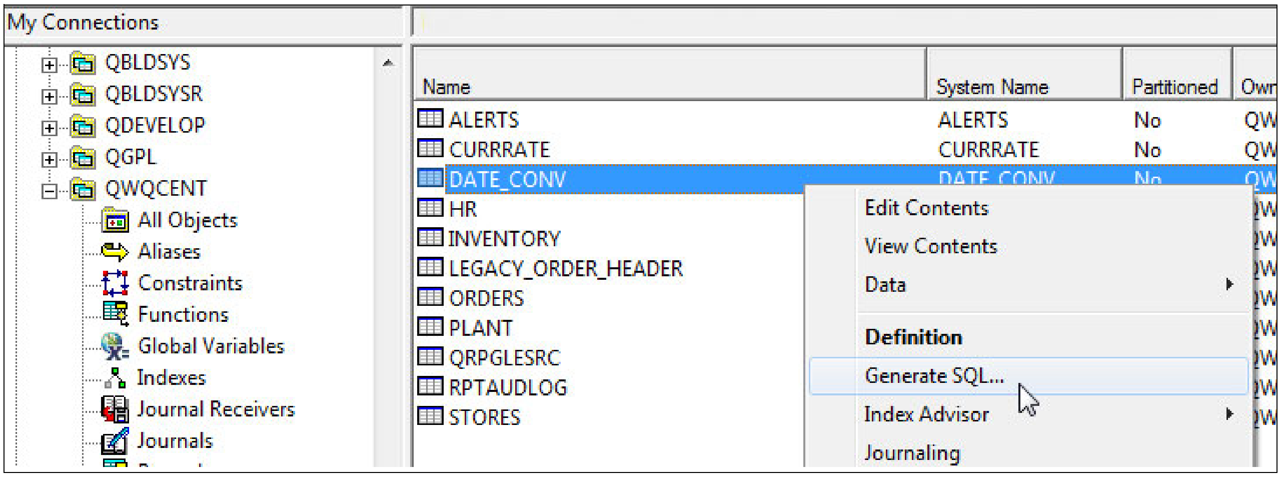


Figure 5-13 Generate SQL for LOAD_DATE_CONVERSION_TABLE procedure

Notice that several of the columns in the date dimension table are *current* flags that contain a value of Y or N depending on whether the condition for that flag is met. For example, the column DC_CURRENT_YEAR contains a Y if that row represents a date whose year is equal to the current year. So, if the current date is June 20, 2012, then the value for the DC_CURRENT_YEAR column is Y for all of the rows that represent dates for the year 2012. Because these are dynamic values, you must add a process to keep these columns in the date dimension table updated. Again a stored procedure that does this is provided in the QWQCENT library and it is named UPDATE_DATE_CONVERSION_TABLE. You can also see the source for this procedure by using the Generate SQL option.

To keep these current flags accurate, all you must do is make sure that the UPDATE_DATE_CONVERSION_TABLE stored procedure runs every day. To do this, create a job scheduler entry to call this stored procedure daily. This can be accomplished by completing the following steps:

1. Create a source file member to store an SQL statement script:

   ```
   ADDPFM FILE(QGPL/QTXTSRC) MBR(UPDDATCONV)
   ```

2. Edit this source file member and add the following SQL statement to call the stored procedure:

   ```
   CALL QWQCENT/UPDATE_DATE_CONVERSION_TABLE
   ```

3. Now, add a job scheduler job to run the SQL script in the source file member:

   ```
   ADDJOBJS JOB(UPDDATCONV) SCDCDE(*DAILY) TIME(0010) CMD(RUNSQLSTM
   SRCFILE(QGPL/QTXTSRC) SRCMBR(UPDDATCONV) COMMIT(*NONE) NAMING(*SYS))
   ```

This job scheduler job runs daily at 1 a.m. and calls the **RUNSQLSTM** command to run the script (in the specified source file member) through SQL.

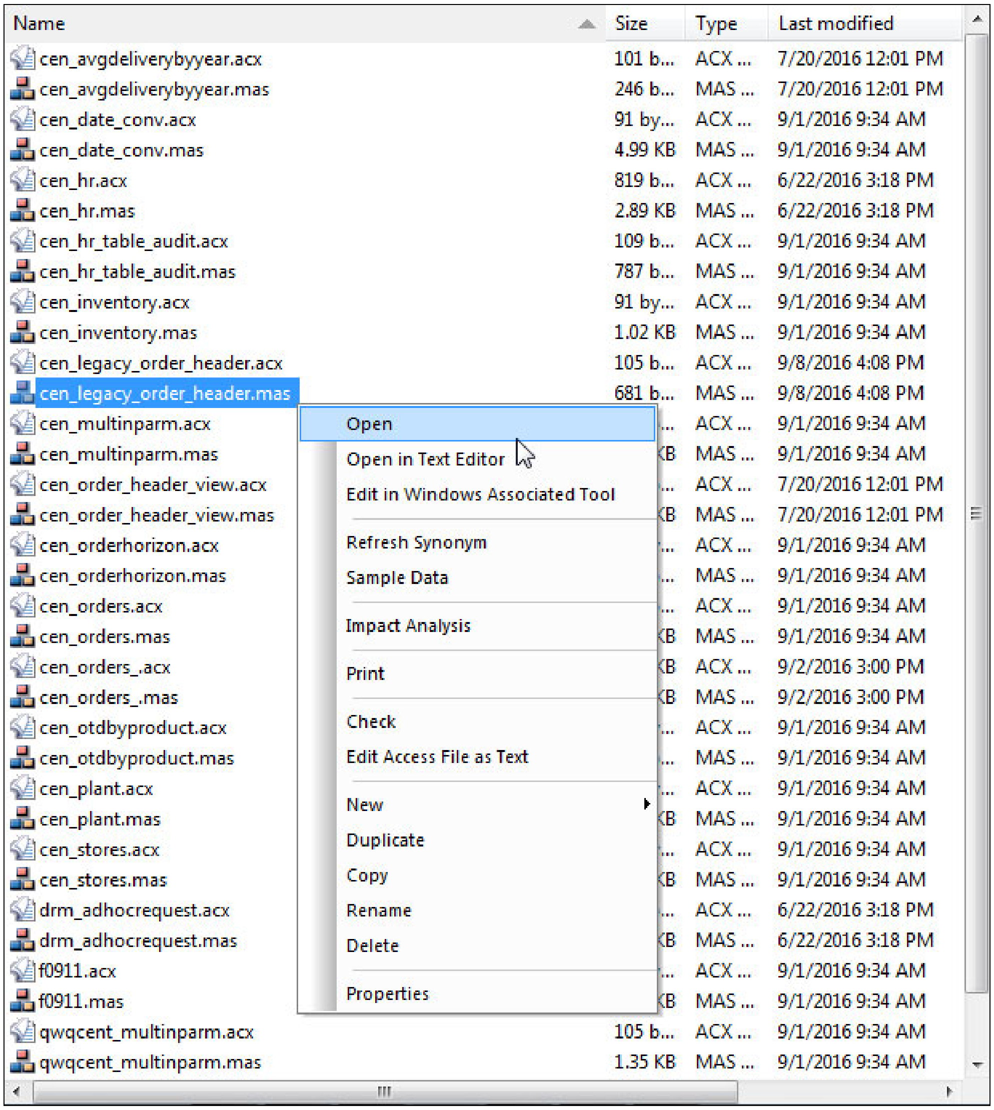
Joining the date dimension table to the table containing your legacy date

There are several methods of joining the date dimension table to the legacy file. Regardless of which method you use, it is generally advantageous to define the joins inside the synonym or in an SQL view rather than within individual reports and graphs for the simple reason that the join logic is centralized in one location. A report developer that uses one of these *pre-joined* synonyms can use any of the fields from the tables that are involved without having to first define the join inside the report or graph.

The following example describes how to implement the join inside the DB2 Web Query for i synonym. You join the synonym for table QWQCENT/LEGACY_ORDER_HEADER (the 10-character system name is ORDHDR) with the synonym for the date dimension table:

1. Create a DB2 Web Query for i synonym over the DATE_CONV table with a prefix of `cen_`, and place it in the `baseapp` application folder.
2. Open DB2 Web Query for i Developer Workbench and open a connection to your environment.
3. Find the `cen_legacy_order_header.mas` file by clicking **Data Servers** → **EDASERVE** → **Applications** → **baseapp**.

4. Right-click this file and select **Open**, as shown in Figure 5-14.



Name	Size	Type	Last modified
cen_avgdeliverybyyear.acx	101 b...	ACX ...	7/20/2016 12:01 PM
cen_avgdeliverybyyear.mas	246 b...	MAS ...	7/20/2016 12:01 PM
cen_date_conv.acx	91 by...	ACX ...	9/1/2016 9:34 AM
cen_date_conv.mas	4.99 KB	MAS ...	9/1/2016 9:34 AM
cen_hr.acx	819 b...	ACX ...	6/22/2016 3:18 PM
cen_hr.mas	2.89 KB	MAS ...	6/22/2016 3:18 PM
cen_hr_table_audit.acx	109 b...	ACX ...	9/1/2016 9:34 AM
cen_hr_table_audit.mas	787 b...	MAS ...	9/1/2016 9:34 AM
cen_inventory.acx	91 by...	ACX ...	9/1/2016 9:34 AM
cen_inventory.mas	1.02 KB	MAS ...	9/1/2016 9:34 AM
cen_legacy_order_header.acx	105 b...	ACX ...	9/8/2016 4:08 PM
cen_legacy_order_header.mas	681 b...	MAS ...	9/8/2016 4:08 PM
cen_multinparm.acx	...	ACX ...	9/1/2016 9:34 AM
cen_multinparm.mas	KB	MAS ...	9/1/2016 9:34 AM
cen_order_header_view.acx	...	ACX ...	7/20/2016 12:01 PM
cen_order_header_view.mas	KB	MAS ...	7/20/2016 12:01 PM
cen_orderhorizon.acx	...	ACX ...	9/1/2016 9:34 AM
cen_orderhorizon.mas	...	MAS ...	9/1/2016 9:34 AM
cen_orders.acx	...	ACX ...	9/1/2016 9:34 AM
cen_orders.mas	KB	MAS ...	9/1/2016 9:34 AM
cen_orders_.acx	KB	ACX ...	9/2/2016 3:00 PM
cen_orders_.mas	KB	MAS ...	9/2/2016 3:00 PM
cen_otdbyproduct.acx	...	ACX ...	9/1/2016 9:34 AM
cen_otdbyproduct.mas	...	MAS ...	9/1/2016 9:34 AM
cen_plant.acx	...	ACX ...	9/1/2016 9:34 AM
cen_plant.mas	KB	MAS ...	9/1/2016 9:34 AM
cen_stores.acx	...	ACX ...	9/1/2016 9:34 AM
cen_stores.mas	KB	MAS ...	9/1/2016 9:34 AM
drm_adhocrequest.acx	...	ACX ...	6/22/2016 3:18 PM
drm_adhocrequest.mas	KB	MAS ...	6/22/2016 3:18 PM
f0911.acx	...	ACX ...	9/1/2016 9:34 AM
f0911.mas	KB	MAS ...	9/1/2016 9:34 AM
qwqcent_multinparm.acx	105 b...	ACX ...	9/1/2016 9:34 AM
qwqcent_multinparm.mas	1.35 KB	MAS ...	9/1/2016 9:34 AM

Figure 5-14 Edit cen_legacy_order_header

At this point, add a join segment to the date dimension table.

5. Right-click the **CEN_LEGACY_ORDER_HEADER** segment name and select **Insert** → **Reference to Existing Synonym**, as shown in Figure 5-15.

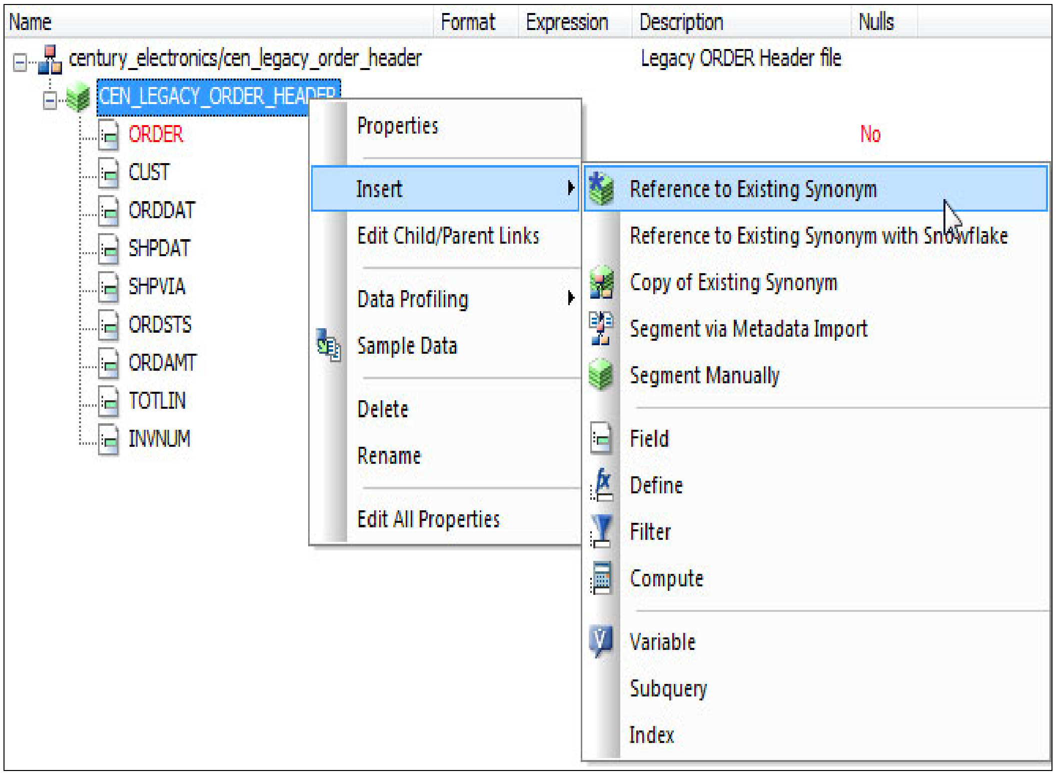
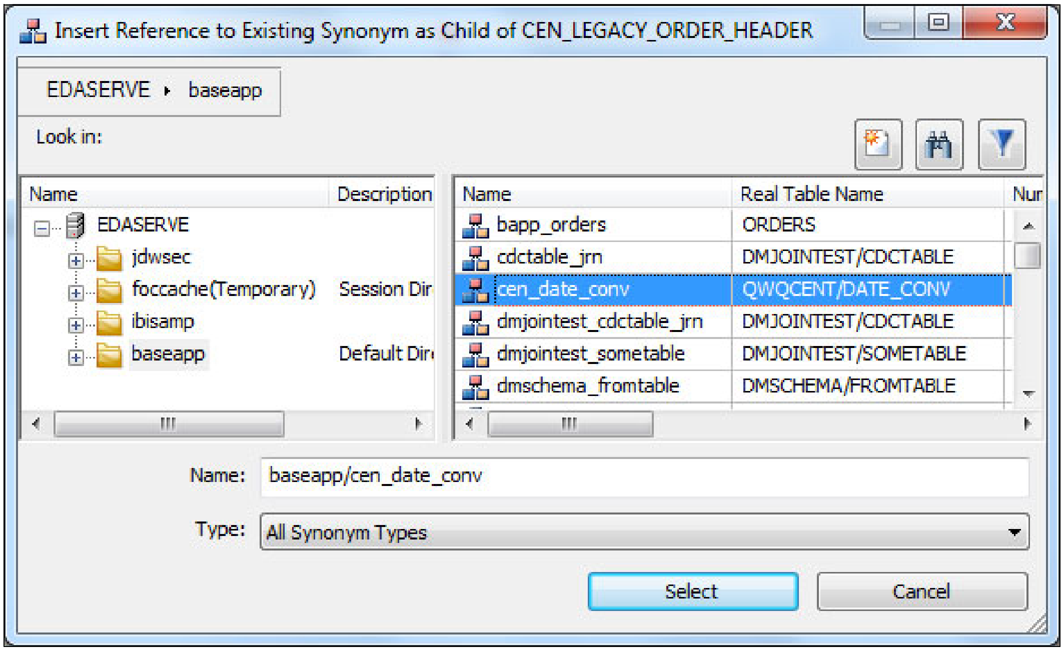


Figure 5-15 Add join segment to legacy_orders_table

A list of synonyms in the BASEAPP folder is presented.

6. As demonstrated in Figure 5-16, find and select your cen_date_conv synonym and click **Select**.



Name	Real Table Name
bapp_orders	ORDERS
cdctable_jrn	DMJOINTEST/CDCTABLE
cen_date_conv	QWQCENT/DATE_CONV
dmjointest_cdctable_jrn	DMJOINTEST/CDCTABLE
dmjointest_sometable	DMJOINTEST/SOMETABLE
dmschema_fromtable	DMSCHEMA/FROMTABLE

Name: baseapp/cen_date_conv

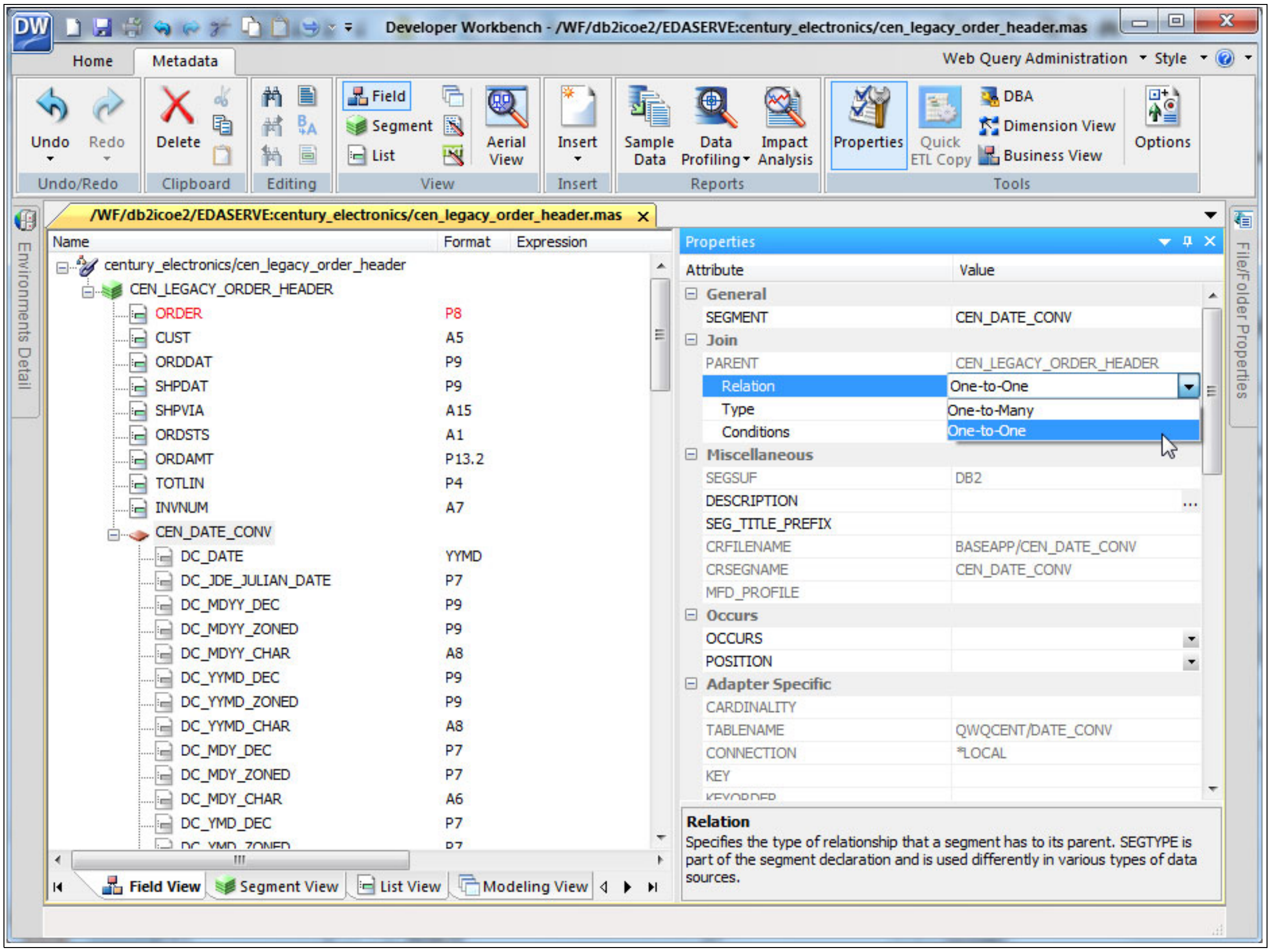
Type: All Synonym Types

Figure 5-16 Select cen_date_conv synonym

The new segment is added to the synonym by reference. Next, you must change the segment type from Multiple to Unique.

7. Select the new **CEN_DATE_CONV** join segment and locate the Relation attribute under the Join section.

8. This is a drop-down list. Click it to and change the Relation to **One-to-One**, as shown in Figure 5-17. This action instructs DB2 Web Query for i to rely on SQL syntax to create the join. One-to-One joins each row from the LEGACY_ORDER_HEADER table to the matching rows from the DATE_CONV table.



Name	Format	Expression
century_electronics/cen_legacy_order_header		
CEN_LEGACY_ORDER_HEADER		
ORDER	P8	
CUST	A5	
ORDDAT	P9	
SHPDAT	P9	
SHPVIA	A15	
ORDSTS	A1	
ORDAMT	P13.2	
TOTLIN	P4	
INVNUM	A7	
CEN_DATE_CONV		
DC_DATE	YYMD	
DC_JDE_JULIAN_DATE	P7	
DC_MDYY_DEC	P9	
DC_MDYY_ZONED	P9	
DC_MDYY_CHAR	A8	
DC_YYMD_DEC	P9	
DC_YYMD_ZONED	P9	
DC_YYMD_CHAR	A8	
DC_MDY_DEC	P7	
DC_MDY_ZONED	P7	
DC_MDY_CHAR	A6	
DC_YMD_DEC	P7	
DC_YMD_ZONED	P7	

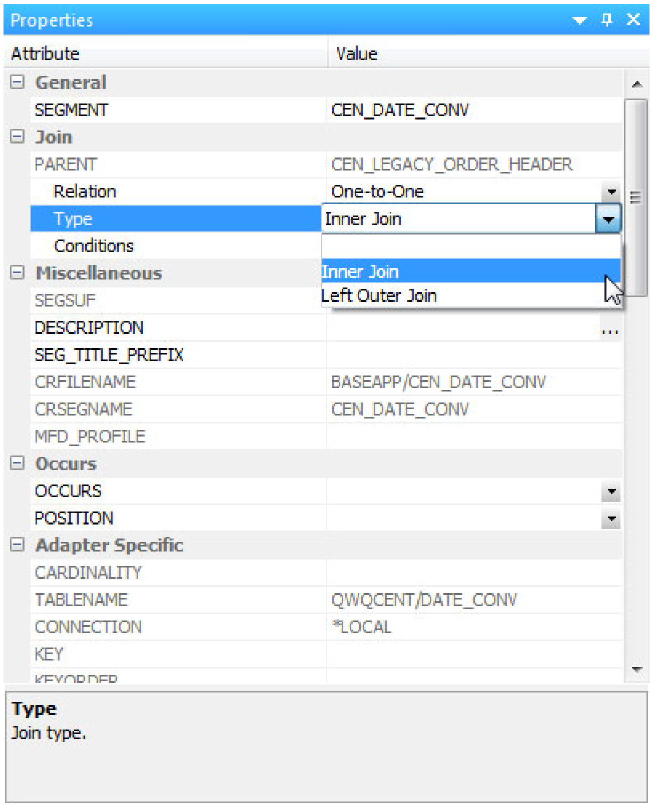
Properties

Attribute	Value
General	
SEGMENT	CEN_DATE_CONV
Join	
PARENT	CEN_LEGACY_ORDER_HEADER
Relation	One-to-One One-to-Many One-to-One
Type	
Conditions	
Miscellaneous	
SEGSUF	DB2
DESCRIPTION	...
SEG_TITLE_PREFIX	
CRFILENAME	BASEAPP/CEN_DATE_CONV
CRSEGNAME	CEN_DATE_CONV
MFD_PROFILE	
Occurs	
OCCURS	
POSITION	
Adapter Specific	
CARDINALITY	
TABLENAME	QWQCENT/DATE_CONV
CONNECTION	*LOCAL
KEY	
KEYORDER	

Relation
Specifies the type of relationship that a segment has to its parent. SEGTYPE is part of the segment declaration and is used differently in various types of data sources.

Figure 5-17 Change the join relationship to One-to-One

9. Locate the Type attribute under the Join section. Click its drop-down list and select Type of **Inner join**, as shown in Figure 5-18. This instructs DB2 Web Query to process this join as an inner join.

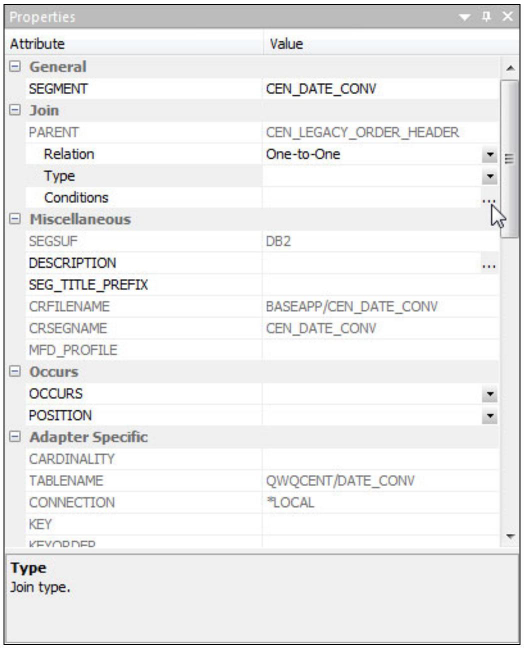


Attribute	Value
General	
SEGMENT	CEN_DATE_CONV
Join	
PARENT	CEN_LEGACY_ORDER_HEADER
Relation	One-to-One
Type	Inner Join
Conditions	
Miscellaneous	
SEGSUF	
DESCRIPTION	
SEG_TITLE_PREFIX	
CRFILENAME	BASEAPP/CEN_DATE_CONV
CRSEGNAME	CEN_DATE_CONV
MFD_PROFILE	
Occurs	
OCCURS	
POSITION	
Adapter Specific	
CARDINALITY	
TABLENAME	QWQCENT/DATE_CONV
CONNECTION	*LOCAL
KEY	
KEYORDER	

Type
Join type.

Figure 5-18 Change the join type to inner join

10. Next, you must define the fields that are used to join the tables together. Locate the Conditions attribute under the Join section. Click **Conditions** (Figure 5-19) to open the Join editor window (Figure 5-20).

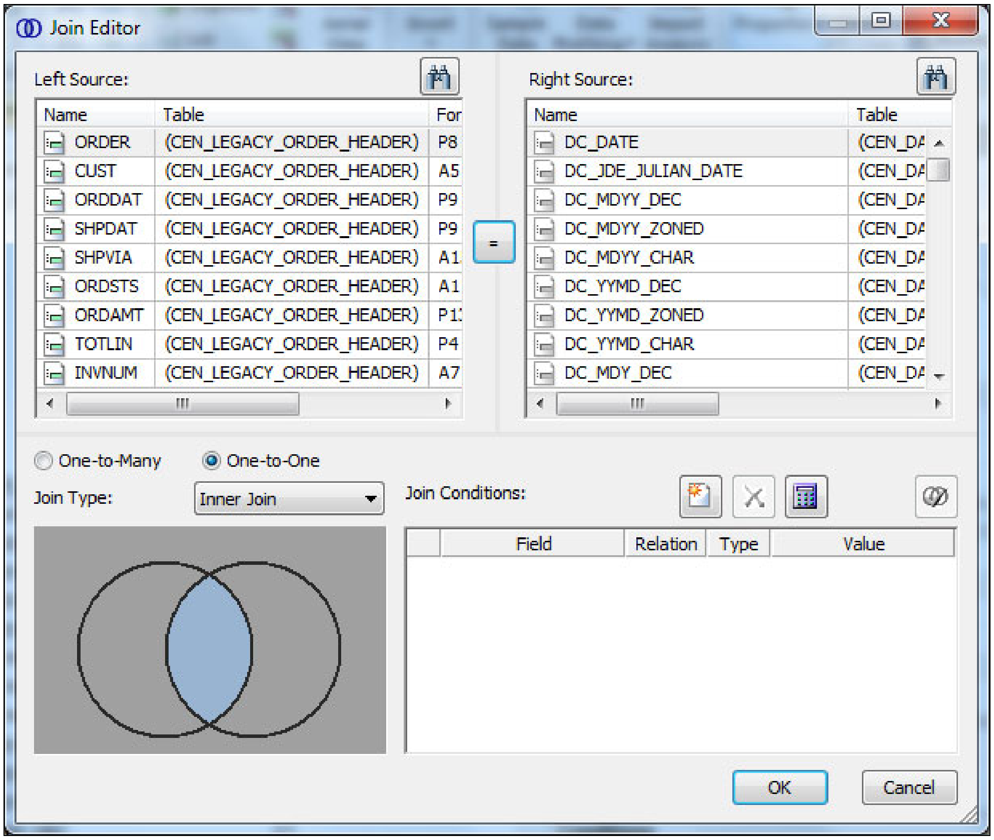


Properties

Attribute	Value
⊟ **General**	
SEGMENT	CEN_DATE_CONV
⊟ **Join**	
PARENT	CEN_LEGACY_ORDER_HEADER
Relation	One-to-One
Type	
Conditions	...
⊟ **Miscellaneous**	
SEGSUF	DB2
DESCRIPTION	...
SEG_TITLE_PREFIX	
CRFILENAME	BASEAPP/CEN_DATE_CONV
CRSEGNAME	CEN_DATE_CONV
MFD_PROFILE	
⊟ **Occurs**	
OCCURS	
POSITION	
⊟ **Adapter Specific**	
CARDINALITY	
TABLENAME	QWQCENT/DATE_CONV
CONNECTION	*LOCAL
KEY	
KEYORDER	

Type
Join type.

Figure 5-19 Open the Join editor window by using the Conditions button



Join Editor

Left Source:

Name	Table	For
ORDER	(CEN_LEGACY_ORDER_HEADER)	P8
CUST	(CEN_LEGACY_ORDER_HEADER)	A5
ORDDAT	(CEN_LEGACY_ORDER_HEADER)	P9
SHPDAT	(CEN_LEGACY_ORDER_HEADER)	P9
SHPVIA	(CEN_LEGACY_ORDER_HEADER)	A1
ORDSTS	(CEN_LEGACY_ORDER_HEADER)	A1
ORDAMT	(CEN_LEGACY_ORDER_HEADER)	P1:
TOTLIN	(CEN_LEGACY_ORDER_HEADER)	P4
INVNUM	(CEN_LEGACY_ORDER_HEADER)	A7

Right Source:

Name	Table
DC_DATE	(CEN_DA
DC_JDE_JULIAN_DATE	(CEN_DA
DC_MDYY_DEC	(CEN_DA
DC_MDYY_ZONED	(CEN_DA
DC_MDYY_CHAR	(CEN_DA
DC_YYMD_DEC	(CEN_DA
DC_YYMD_ZONED	(CEN_DA
DC_YYMD_CHAR	(CEN_DA
DC_MDY_DEC	(CEN_DA

○ One-to-Many ● One-to-One

Join Type: Inner Join

Join Conditions:

Field	Relation	Type	Value

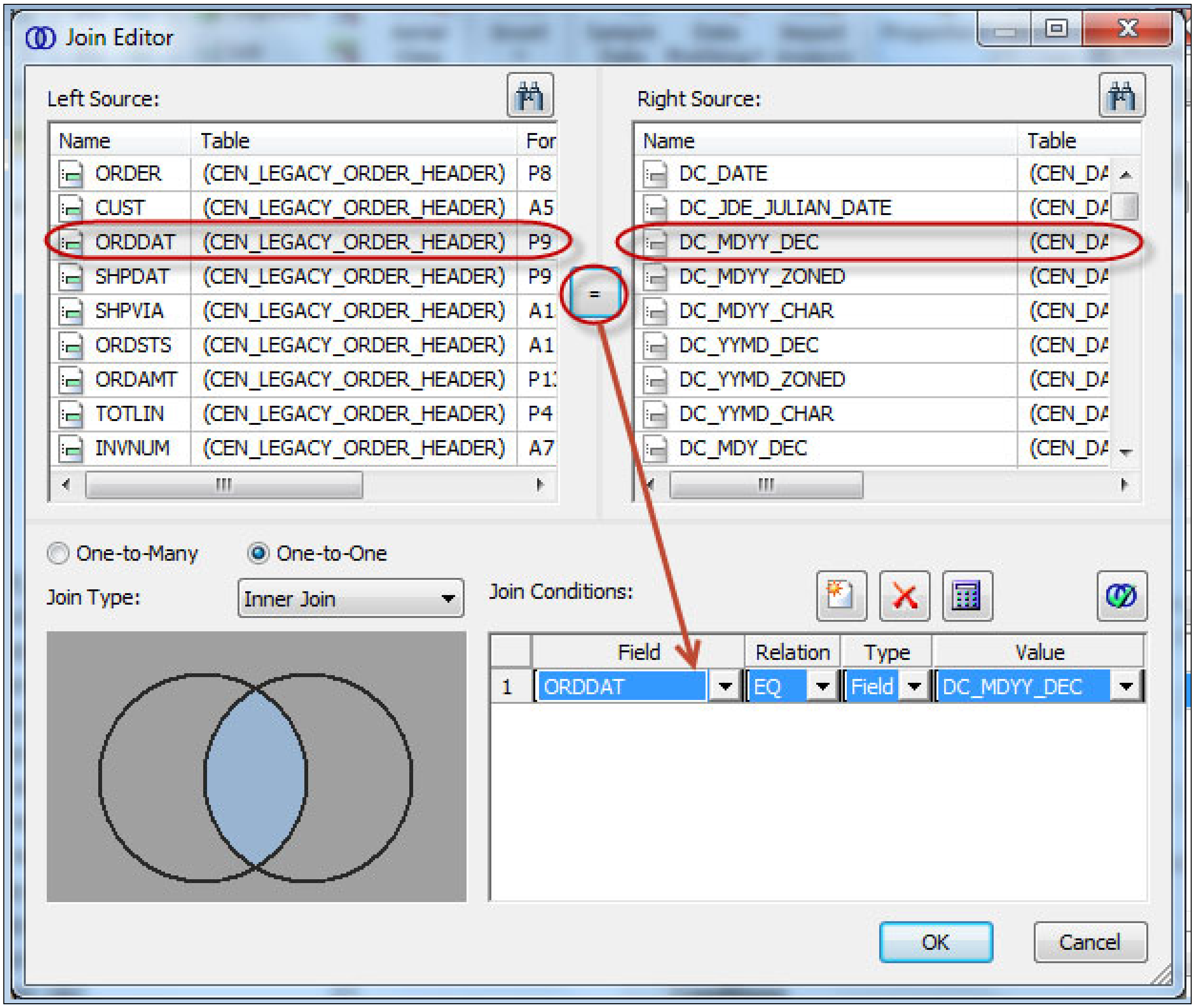
Figure 5-20 Join Editor window

11. In the Join Editor window, complete the following steps:

   a. From the list of columns in the CEN_LEGACY_ORDER_HEADER segment, select **ORDDAT** so that it is highlighted.

   b. From the list of columns in the CEN_DATE_CONV segment, select **DC_MDYY_DEC** so that it is highlighted. The values in the ORDDAT are in this format, so this is the column in the date dimension table to use as the join column.

   c. Click the **=** icon that is located between the two segment lists. This action should bring the selected join columns down into the expression pane. An example is shown in Figure 5-21.



Left Source:

Name	Table	For
ORDER	(CEN_LEGACY_ORDER_HEADER)	P8
CUST	(CEN_LEGACY_ORDER_HEADER)	A5
ORDDAT	(CEN_LEGACY_ORDER_HEADER)	P9
SHPDAT	(CEN_LEGACY_ORDER_HEADER)	P9
SHPVIA	(CEN_LEGACY_ORDER_HEADER)	A1
ORDSTS	(CEN_LEGACY_ORDER_HEADER)	A1
ORDAMT	(CEN_LEGACY_ORDER_HEADER)	P1:
TOTLIN	(CEN_LEGACY_ORDER_HEADER)	P4
INVNUM	(CEN_LEGACY_ORDER_HEADER)	A7

Right Source:

Name	Table
DC_DATE	(CEN_DA
DC_JDE_JULIAN_DATE	(CEN_DA
DC_MDYY_DEC	(CEN_DA
DC_MDYY_ZONED	(CEN_DA
DC_MDYY_CHAR	(CEN_DA
DC_YYMD_DEC	(CEN_DA
DC_YYMD_ZONED	(CEN_DA
DC_YYMD_CHAR	(CEN_DA
DC_MDY_DEC	(CEN_DA

○ One-to-Many ● One-to-One

Join Type: Inner Join

Join Conditions:

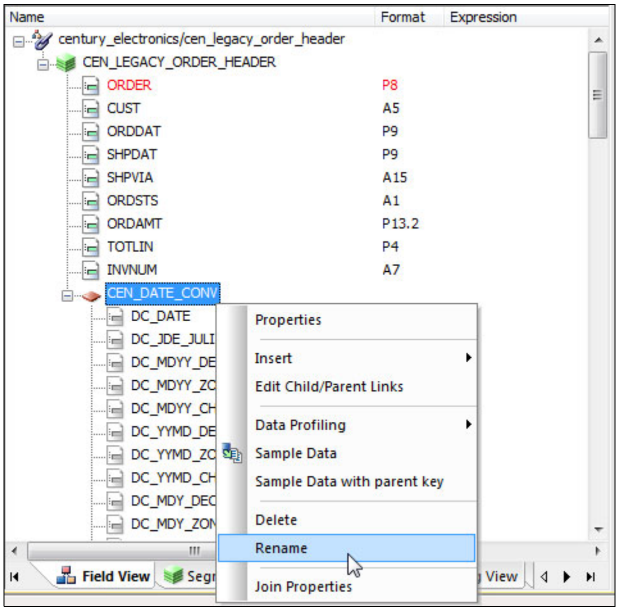
	Field	Relation	Type	Value
1	ORDDAT	EQ	Field	DC_MDYY_DEC

Figure 5-21 Define join columns

12. Click **OK**. To help the report developers identify what legacy date field this date dimension segment represents, rename the segment to the name of the field that it is converting. If you do not do this, the segment names to the date dimension table might be ambiguous (your report developers might not be certain of what legacy date field they represent). This is particularly true if you have multiple legacy date fields in your synonym because to convert them you must define a join segment to date_conv for each one.

> **Tip:** If you are meticulous (and ambitious), you can remove the ambiguity by renaming each of the individual column names in the date dimension segments (for example, rename DC_YEAR to ORDDAT_YEAR). This can be done by highlighting the column, right-clicking, and selecting **Rename**.

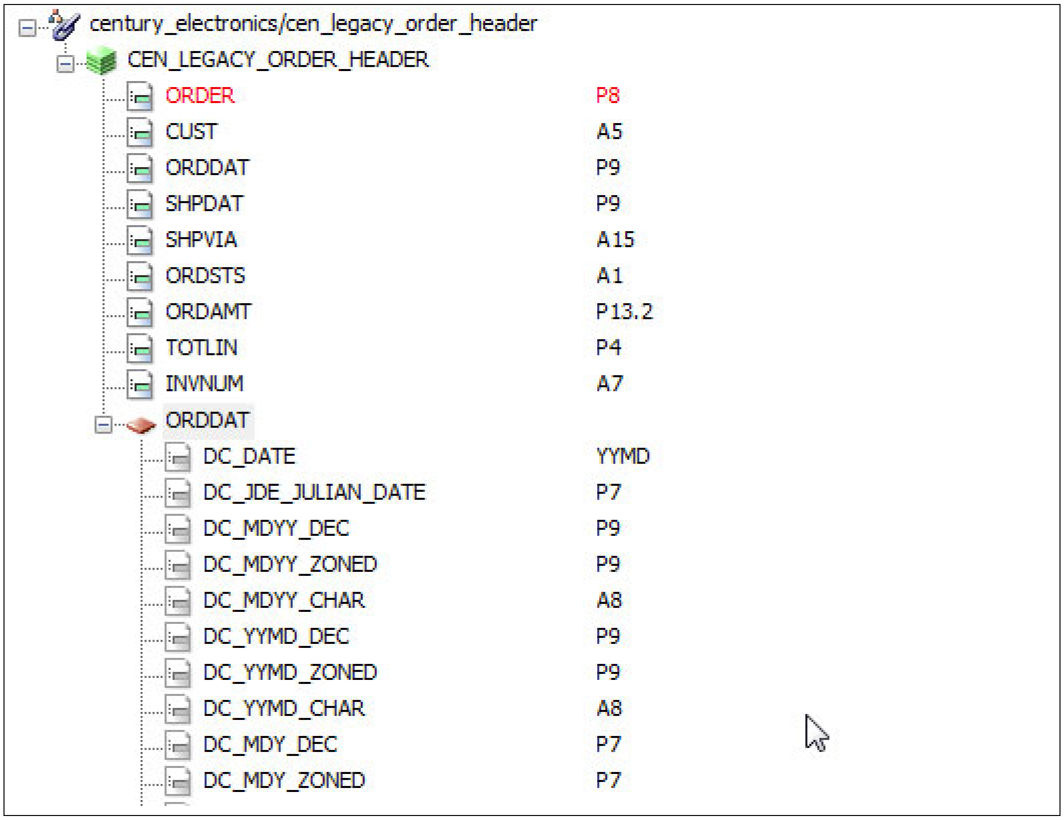
13. Highlight the segment, right-click it, and select **Rename** from the right-click menu. This is shown in Figure 5-22.



Name	Format	Expression
century_electronics/cen_legacy_order_header		
CEN_LEGACY_ORDER_HEADER		
ORDER	P8	
CUST	A5	
ORDDAT	P9	
SHPDAT	P9	
SHPVIA	A15	
ORDSTS	A1	
ORDAMT	P13.2	
TOTLIN	P4	
INVNUM	A7	
CEN_DATE_CONV		
DC_DATE		
DC_JDE_JULI		
DC_MDYY_DE		
DC_MDYY_ZC		
DC_MDYY_CH		
DC_YYMD_DE		
DC_YYMD_ZC		
DC_YYMD_CH		
DC_MDY_DEC		
DC_MDY_ZON		

Figure 5-22 Rename the segment

14. Specify **ORDDAT** as the segment name. When you are finished, your synonym looks like the example that is provided in Figure 5-23.



```
century_electronics/cen_legacy_order_header
  CEN_LEGACY_ORDER_HEADER
    ORDER                     P8
    CUST                      A5
    ORDDAT                    P9
    SHPDAT                    P9
    SHPVIA                    A15
    ORDSTS                    A1
    ORDAMT                    P13.2
    TOTLIN                    P4
    INVNUM                    A7
    ORDDAT
      DC_DATE                 YYMD
      DC_JDE_JULIAN_DATE      P7
      DC_MDYY_DEC             P9
      DC_MDYY_ZONED           P9
      DC_MDYY_CHAR            A8
      DC_YYMD_DEC             P9
      DC_YYMD_ZONED           P9
      DC_YYMD_CHAR            A8
      DC_MDY_DEC              P7
      DC_MDY_ZONED            P7
```

Figure 5-23 Synonym with new ORDDAT segment

15. Repeat steps 5 on page 89 - 14 for the SHPDAT legacy date field. Be sure that you rename the segment SHPDAT.

16. Save the synonym.

17. Close the Edit synonym window.

> **Tip:** For performance reasons, now is a good time to create indexes over your join columns. The DB2 optimizer can use these indexes for statistics and implementation during query execution. Here are the two SQL statements for creating indexes over the LEGACY_ORDER_HEADER and DATE_CONV tables:
>
> ```
> CREATE INDEX LEGACY_ORDER_HEADER_INDEX_00001 ON LEGACY_ORDER_HEADER (ORDDAT
> ASC);
> CREATE INDEX DATE_CONV_INDEX_00001 ON DATE_CONV (DC_MDYY_DEC  ASC);
> ```

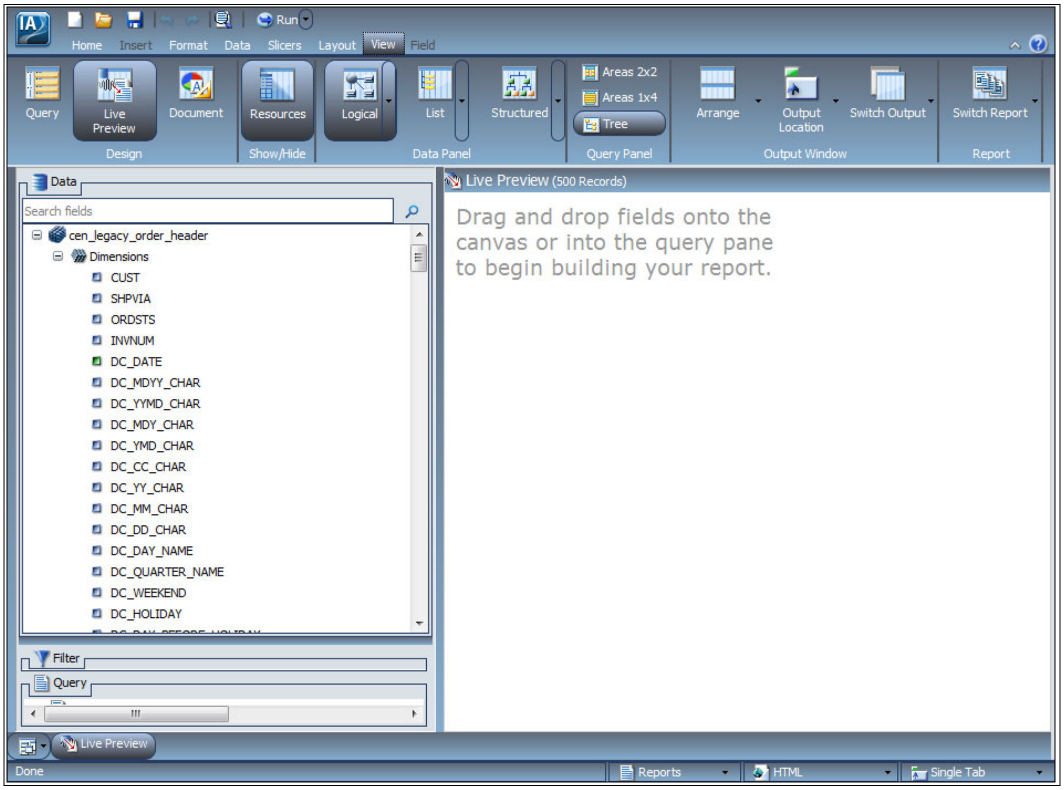
Creating reports by using the date dimension table

After the join is defined in the synonym, the next step is to create a report over this updated synonym. Complete the following steps:

1. Return to your DB2 Web Query browser session.
2. Create a report in the Century Electronics folder by using InfoAssist.

3. Select the **CEN_LEGACY_ORDER_HEADER** synonym as the report data source.

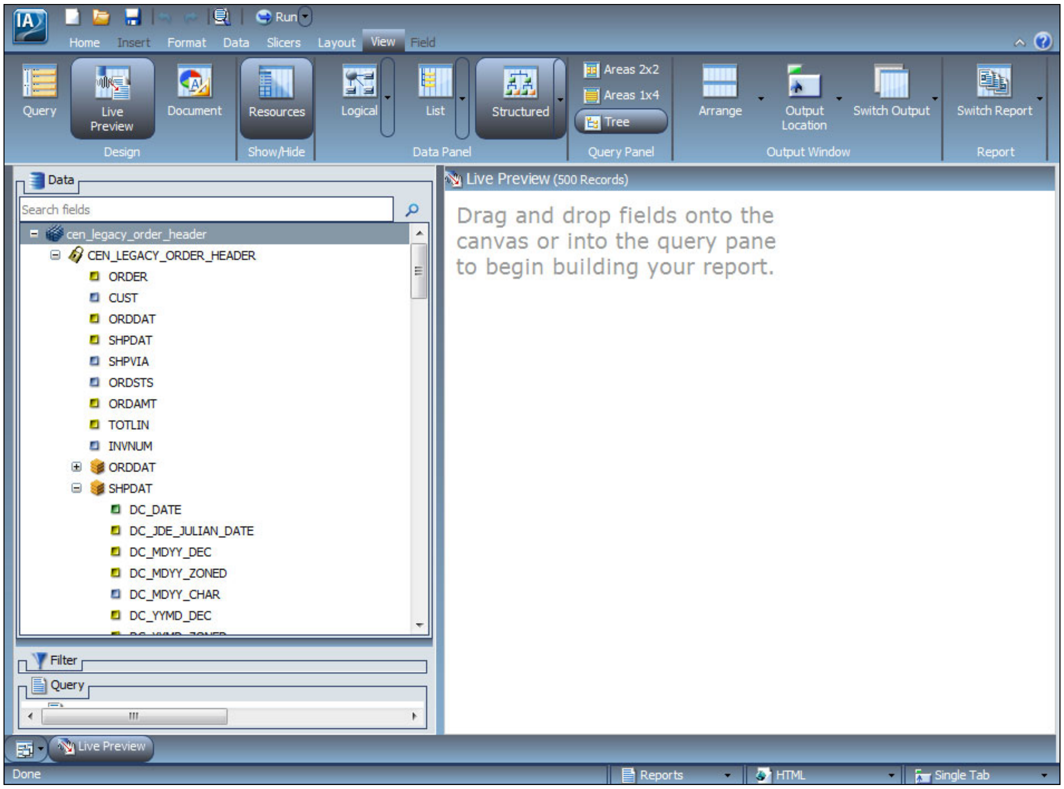
   The InfoAssist tool is presented. Notice that all of the columns from the date dimension table now appear in the list of fields to choose from. If your default view is set to Logical, as shown in Figure 5-24, you might notice that the list is now long and cluttered.



Drag and drop fields onto the canvas or into the query pane to begin building your report.

Figure 5-24 InfoAssist in Logical Data window view

4. To organize the list so that you know which date fields are coming from which joins, click the Structured icon under the View ribbon. This organizes the list into a structured hierarchical tree view, as shown in Figure 5-25. This action groups the synonym columns under their respective segments and allows you to expand and collapse the list of columns under each segment.



Drag and drop fields onto the canvas or into the query pane to begin building your report.

Figure 5-25 Switching Data Panel to Structured view

5. Finish the report by completing the following steps:

   a. Drag the new ORDDAT.DC_YEAR field into the By field container in the Query pane.

   b. Drag the ORDDAT.DC_DOW field into the Across field container in the Query pane. Make it an invisible field, as shown in Figure 5-26. DC_DOW is the numeric value of the day of the week. You must sort by this field, but not by the name of the day of the week field (DC_DAY_NAME). Otherwise, the data for Friday would appear first (since it is the first day name alphabetically) and Wednesday would appear last. However, you probably do not want the numeric day of the week value to actually appear on the report, so hide this column in the report.

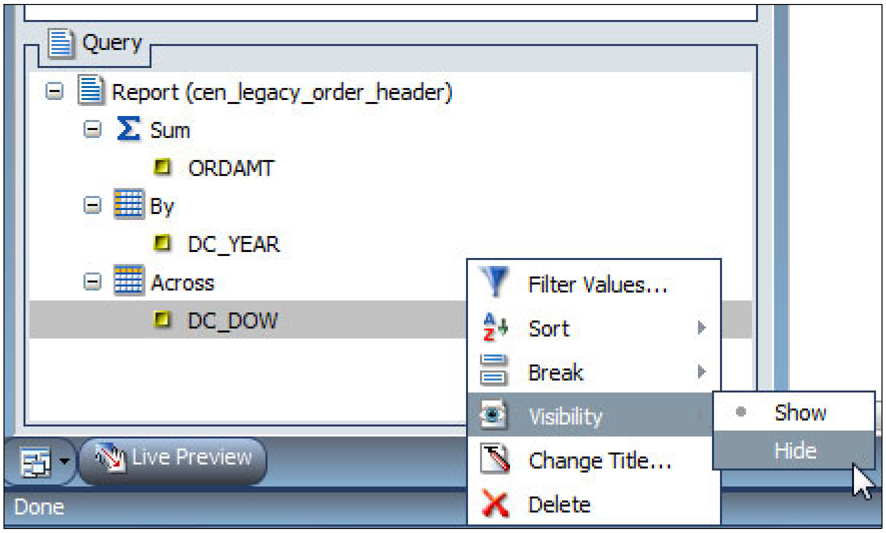


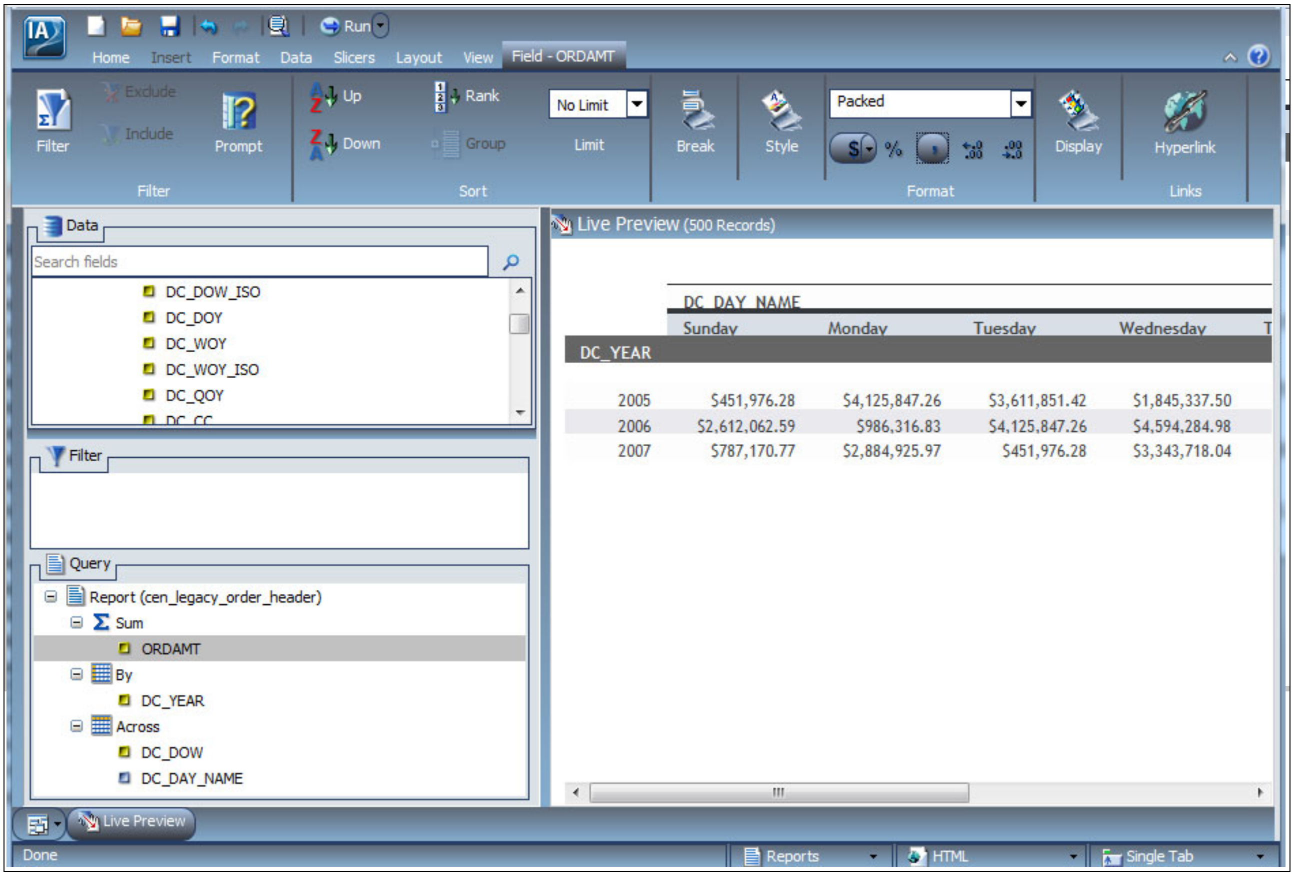
Figure 5-26 Hide the DC_DOW column

   c. Drag the ORDDAT.DC_DAY_NAME field into the Across field container in the Query pane. Make sure it is after DC_DOW.

   d. Drag the ORDDAT.ORDAMT field into the Sum field container in the Query pane and select it. Click the **Show field options** icon for this field.

   e. Add comma inclusion and floating currency formatting to the ORDAMT field in the report.

When finished, the report definition should look like the example that is provided in Figure 5-27.



- DC_DOW_ISO
- DC_DOY
- DC_WOY
- DC_WOY_ISO
- DC_QOY
- DC_CC

Query:

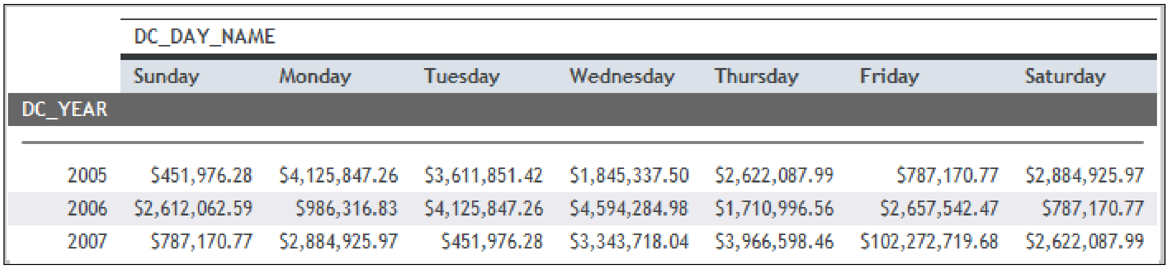
- Report (cen_legacy_order_header)
  - Sum
    - ORDAMT
  - By
    - DC_YEAR
  - Across
    - DC_DOW
    - DC_DAY_NAME

Live Preview (500 Records)

	DC_DAY_NAME				
	Sunday	Monday	Tuesday	Wednesday	T
DC_YEAR					
2005	$451,976.28	$4,125,847.26	$3,611,851.42	$1,845,337.50	
2006	$2,612,062.59	$986,316.83	$4,125,847.26	$4,594,284.98	
2007	$787,170.77	$2,884,925.97	$451,976.28	$3,343,718.04	

Figure 5-27 Report that uses the date dimension table

6. Run the report. The results should look identical to those in Figure 5-28.



	DC_DAY_NAME						
	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
DC_YEAR							
2005	$451,976.28	$4,125,847.26	$3,611,851.42	$1,845,337.50	$2,622,087.99	$787,170.77	$2,884,925.97
2006	$2,612,062.59	$986,316.83	$4,125,847.26	$4,594,284.98	$1,710,996.56	$2,657,542.47	$787,170.77
2007	$787,170.77	$2,884,925.97	$451,976.28	$3,343,718.04	$3,966,598.46	$102,272,719.68	$2,622,087.99

Figure 5-28 Results of report that uses the date dimension table

Other than the fields and techniques that are used to derive the true dates from legacy dates, this is the same report that you created in 5.2.2, "Using DB2 Web Query for i functions to convert to smart dates" on page 72. Compare the results of those two reports. They should be identical.

7. Save your report as `Date conversion using date dimension table`.

> **Tip:** Now is an opportune time to create indexes over the join columns.

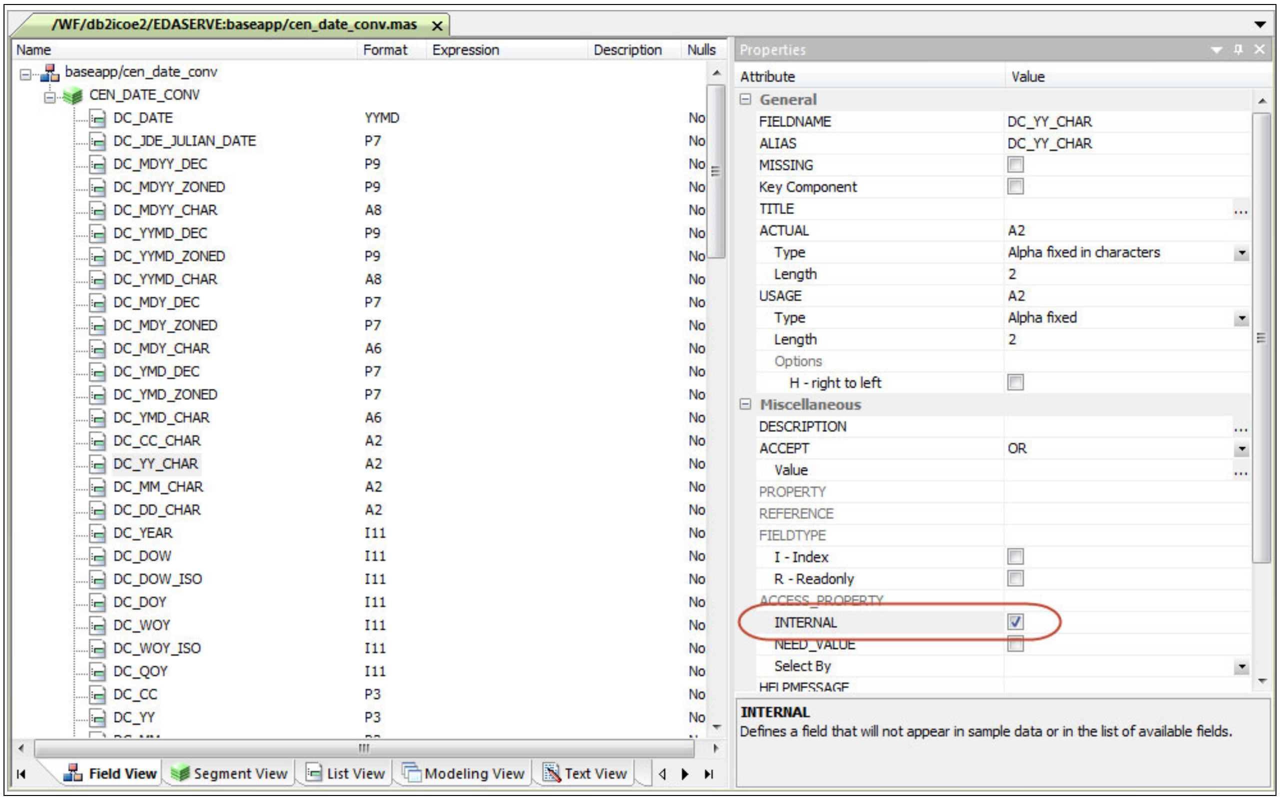
One of the advantages of the date dimension table method is that all of the columns in the table are brought into the synonyms segment and subsequently appear in the list of available fields in the development tools. This gives report developers the ability to easily and effortlessly include a wide variety of date formats in your reports.

One of the disadvantages is that all of the columns in the table appear in the list of available fields in the development tools. Why is this a disadvantage? Because there might be many columns in the table and some of them might never be used in the reports. These columns can clutter up the list, particularly if the synonym contains many other columns.

Furthermore, if you have multiple legacy date fields in your file and you want to convert them to date fields, you must create join segments to the date dimension table for each one. This adds even more columns to the list of fields in the development tools.

Here are some suggestions for keeping the list of available fields to a manageable size:

- Create the date dimension table with fewer columns (only the ones that your use regularly).
- Use a DB2 view to join the needed tables, including only the columns from each needed for reporting. The resulting synonym that is created over this view then contains only the columns that are useful for reporting purposes.
- Hide unneeded columns within the synonym by setting their column Access Property to INTERNAL so that you can preserve the columns in the date dimension table while not showing them in the development tools. If they are needed for some reports, you can simply unhide them by clearing the INTERNAL setting. Figure 5-29 provides an example of how to hide DC_MONTH_NAME so that it does not appear in the list of available columns in the report development tools.



Name	Format	Expression	Description	Nulls
baseapp/cen_date_conv				
CEN_DATE_CONV				
DC_DATE	YYMD			No
DC_JDE_JULIAN_DATE	P7			No
DC_MDYY_DEC	P9			No
DC_MDYY_ZONED	P9			No
DC_MDYY_CHAR	A8			No
DC_YYMD_DEC	P9			No
DC_YYMD_ZONED	P9			No
DC_YYMD_CHAR	A8			No
DC_MDY_DEC	P7			No
DC_MDY_ZONED	P7			No
DC_MDY_CHAR	A6			No
DC_YMD_DEC	P7			No
DC_YMD_ZONED	P7			No
DC_YMD_CHAR	A6			No
DC_CC_CHAR	A2			No
DC_YY_CHAR	A2			No
DC_MM_CHAR	A2			No
DC_DD_CHAR	A2			No
DC_YEAR	I11			No
DC_DOW	I11			No
DC_DOW_ISO	I11			No
DC_DOY	I11			No
DC_WOY	I11			No
DC_WOY_ISO	I11			No
DC_QOY	I11			No
DC_CC	P3			No
DC_YY	P3			No

Properties

Attribute	Value
General	
FIELDNAME	DC_YY_CHAR
ALIAS	DC_YY_CHAR
MISSING	□
Key Component	□
TITLE	...
ACTUAL	A2
Type	Alpha fixed in characters
Length	2
USAGE	A2
Type	Alpha fixed
Length	2
Options	
H - right to left	□
Miscellaneous	
DESCRIPTION	...
ACCEPT	OR
Value	...
PROPERTY	
REFERENCE	
FIELDTYPE	
I - Index	□
R - Readonly	□
ACCESS_PROPERTY	
INTERNAL	☑
NEED_VALUE	□
Select By	
HELPMESSAGE	

INTERNAL
Defines a field that will not appear in sample data or in the list of available fields.

Figure 5-29 Define a column that does not appear in the list of available columns

Attention: Existing reports that reference hidden columns (those whose INTERNAL setting for ACCESS PROPERTY is selected) continue to run. However, if you attempt to open such a report in the report development tools, the request fails with a `Field not found in master file` error. In those cases, you must edit the synonym and remove (clear) the INTERNAL setting for ACCESS PROPERTY.

5.2.5 Comparing the legacy date conversion techniques

At this point, you might be wondering whether it matters what legacy date conversion technique you implement. From a functional standpoint, the answer is no. All three methods provide reliable date conversion results. However, from a report-performance perspective, the answer is yes. Pushing the date conversion logic down to the database engine provides significant report performance advantages, but exactly to what degree always varies, depending on factors such as system resources, system activity, available indexes, the number of rows retrieved, and the number of columns converted. But, to give you some idea of what gains can potentially be realized, a mini-benchmark was found to compare the performance results of the two techniques. The benchmark used a 6,000,000 row ORDERS table. The reports that were used were essentially identical to the two that you created in the above date conversion exercise.

Each version of the report was run 20 times, and the run time (the time between when the user selected the report and when the report was displayed to the browser) was recorded and averaged. The results are listed in Table 5-2.

Table 5-2 Results

Date conversion technique	Average run time (20 executions)
DB2 Web Query for i functions (BIFs)	1 minute, 55 seconds
Date dimension table	5.2 seconds
SQL functions in an SQL view	5.2 seconds

The reason why the date dimension table and SQL function technique perform so much better than the BIFs is because the conversion logic processing is pushed down to the DB2 engine. However, these are two vastly different DB2 approaches:

- The date dimension method performs database joins to locate the row with the date attribute information that is already converted.
- The SQL function method performs date attribute conversions dynamically.

Because the DB2 SQL Query Engine (SQE) processes the requests, both can cache the results so that subsequent requests might perform better. As you can see in the foregoing benchmark results, the date dimension table and SQL functions in an SQL view had similar performance.

Performing the conversion in SQL views and functions might require a fairly elaborate view definition and does require some degree of SQL knowledge to implement. Conversely, the date dimension table can be implemented by someone with limited SQL skills. The SQL views that are created with either of these methods can be used outside of DB2 Web Query.

Performance notwithstanding, the best technique is somewhat subjective. For simplicity, many report developers start out by using the DB2 Web Query for i BIFs, and if the date attribute conversion requirements are simple or the data sets involved are relatively small, the response time of the BIFs might provide acceptable performance. However, if the performance is not satisfactory, then you must strongly consider using one of the other two DB2-based techniques.

When deciding between the date dimension table or SQL function/view method, most people likely find the date dimension table method to be a bit more flexible and easier to implement. Because the information is in a table that can be viewed and maintained, it is easier for you to understand and comprehend. Implementing customized columns such as fiscal year/quarter, season and a Super Bowl Sunday flag is also easier to manage because you can manually set these values in the table and view the data. The date dimension table is infinitely customizable and extendable; after all, there are no SEASON or SUPERBOWLSUNDAY SQL functions.

5.2.6 Example: Converting date fields in the Oracle JD Edwards World application

The Oracle JD Edwards World application has a large installation base on IBM i. The database model for this application stores dates in a specific nonstandard format. Therefore, many Oracle JD Edwards World customers that use DB2 Web Query must know how to handle this specific type of date conversion.

Date fields in the Oracle JD Edwards World application are typically defined as ZONED (6,0) and are in the format CYYDDD broken down as follows:

- C represents the century indicator:
  - 0 for any date before January 2000
  - 1 for any date on or after January 1, 2000
- YY represents the 2-digit year. Dates for the year 2012 are stored as 12.
- DDD represents the number of the day of the year, counting from January 1. So for August 27, 2012, these digits contain the value 239.

  For August 27, 2012, the full value for this field is 112239 and for August 27, 1999, it is 009239.

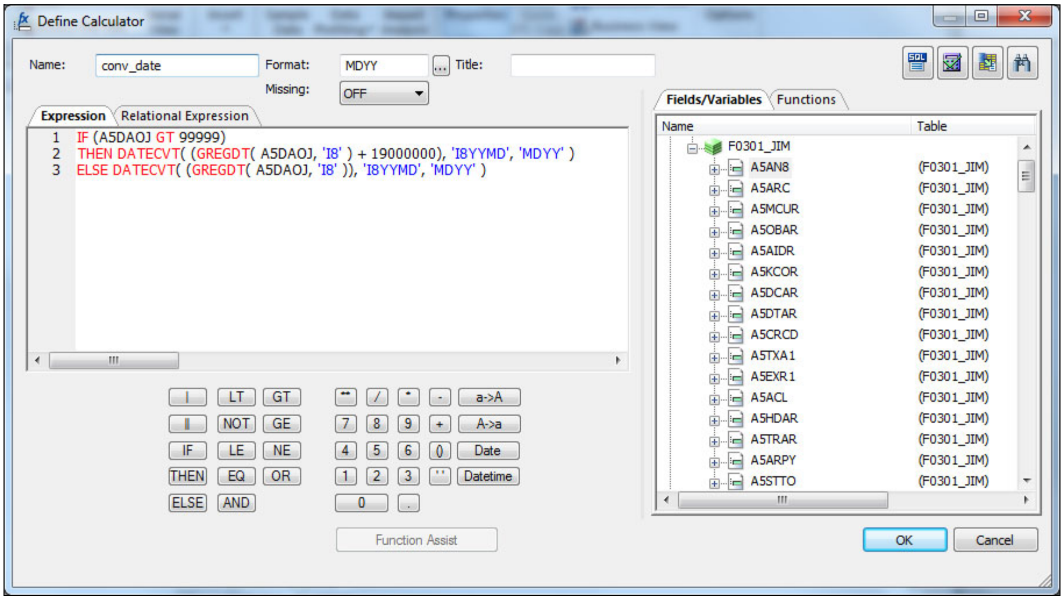
To convert this field to a true date, use one of the following approaches (assume that the Oracle JD Edwards World file is `F0301` and the date field is A5DAOJ):

- DB2 Web Query BIF:

  Use the following expression (where A5DAOJ is the World date field):

```
IF (A5DAOJ GT 99999) THEN
DATECVT( (GREGDT( A5DAOJ, 'I8' ) + 19000000), 'I8YYMD', 'MDYY' )
ELSE DATECVT( (GREGDT( A5DAOJ, 'I8' )), 'I8YYMD', 'MDYY' )
```

  An example is shown in Figure 5-30.



Name: conv_date
Format: MDYY
Title:
Missing: OFF

```
IF (A5DAOJ GT 99999)
THEN DATECVT( (GREGDT( A5DAOJ, 'I8' ) + 19000000), 'I8YYMD', 'MDYY' )
ELSE DATECVT( (GREGDT( A5DAOJ, 'I8' )), 'I8YYMD', 'MDYY' )
```

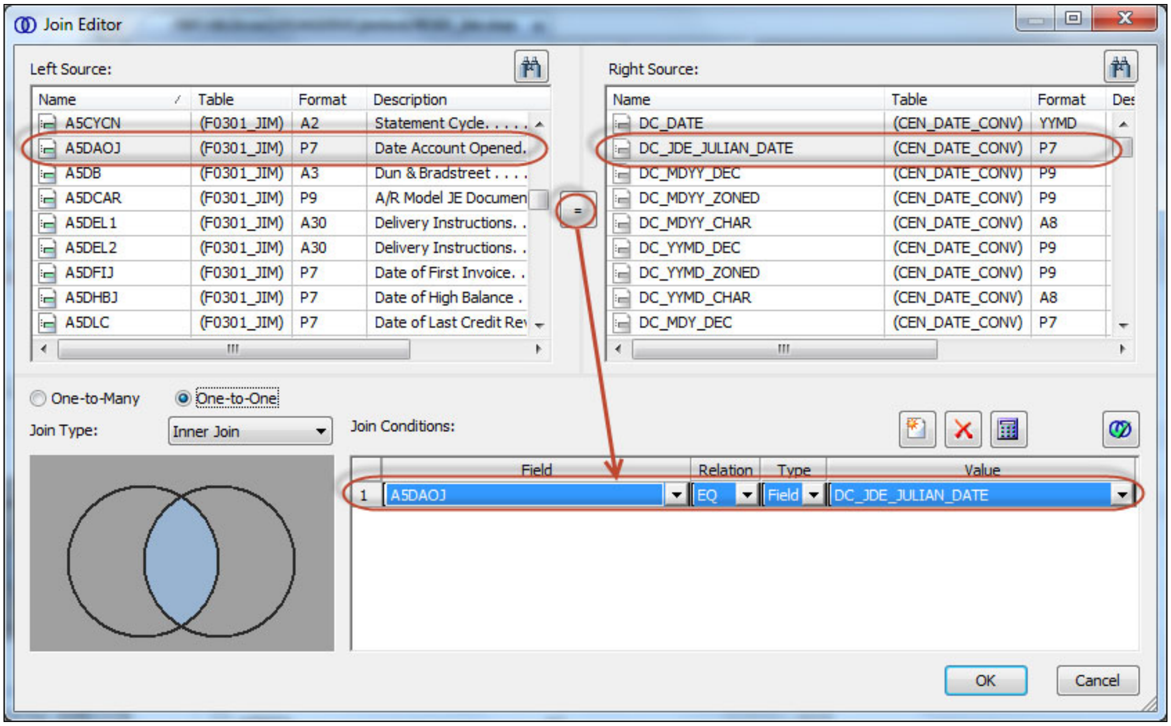
Name	Table
F0301_JIM	
A5AN8	(F0301_JIM)
A5ARC	(F0301_JIM)
A5MCUR	(F0301_JIM)
A5OBAR	(F0301_JIM)
A5AIDR	(F0301_JIM)
A5KCOR	(F0301_JIM)
A5DCAR	(F0301_JIM)
A5DTAR	(F0301_JIM)
A5CRCD	(F0301_JIM)
A5TXA1	(F0301_JIM)
A5EXR1	(F0301_JIM)
A5ACL	(F0301_JIM)
A5HDAR	(F0301_JIM)
A5TRAR	(F0301_JIM)
A5ARPY	(F0301_JIM)
A5STTO	(F0301_JIM)

Figure 5-30 Converting the Oracle JD Edwards World date

- Date dimension table:

  To implement the date dimension table method for Oracle JD Edwards World date fields, create a join from the Oracle JD Edwards World file to the date dimension table by using the Oracle JD Edwards World date field (in this case A5DAOJ) and `DC_JDE_JULIAN_DATE` as the join columns.

Figure 5-31 shows an example of how to define this join in the synonym.



Name	Table	Format	Description
A5CYCN	(F0301_JIM)	A2	Statement Cycle. . . . .
A5DAOJ	(F0301_JIM)	P7	Date Account Opened.
A5DB	(F0301_JIM)	A3	Dun & Bradstreet . . . .
A5DCAR	(F0301_JIM)	P9	A/R Model JE Documen
A5DEL1	(F0301_JIM)	A30	Delivery Instructions. .
A5DEL2	(F0301_JIM)	A30	Delivery Instructions. .
A5DFIJ	(F0301_JIM)	P7	Date of First Invoice. .
A5DHBJ	(F0301_JIM)	P7	Date of High Balance .
A5DLC	(F0301_JIM)	P7	Date of Last Credit Rev

Name	Table	Format	Des
DC_DATE	(CEN_DATE_CONV)	YYMD	
DC_JDE_JULIAN_DATE	(CEN_DATE_CONV)	P7	
DC_MDYY_DEC	(CEN_DATE_CONV)	P9	
DC_MDYY_ZONED	(CEN_DATE_CONV)	P9	
DC_MDYY_CHAR	(CEN_DATE_CONV)	A8	
DC_YYMD_DEC	(CEN_DATE_CONV)	P9	
DC_YYMD_ZONED	(CEN_DATE_CONV)	P9	
DC_YYMD_CHAR	(CEN_DATE_CONV)	A8	
DC_MDY_DEC	(CEN_DATE_CONV)	P7	

○ One-to-Many ● One-to-One

Join Type: Inner Join

Join Conditions:

	Field	Relation	Type	Value
1	A5DAOJ	EQ	Field	DC_JDE_JULIAN_DATE

Figure 5-31 Using the date dimension table to convert the Oracle JD Edwards World date

- SQL functions/view

  If you prefer the SQL functions and view technique, use something like the following code to convert the field to a date:

  ```
  SELECT DATE(DIGITS( DECIMAL(pccefj + 1900000 , 7 ,0))) AS conv_date FROM f43090
  ```

5.2.7 Converting other common legacy date formats

IBM i customers have represented their date fields in many different formats over the years. For each of the three techniques, some of the more common formats and how to convert them, are provided in Table 5-3.

Table 5-3 Legacy date conversion table

Example legacy date field name	Legacy data type	Legacy date format	Legacy example (August 31, 2012)	DB2 Web Query BIF Expression	Date dimension table	SQL function
CH_MDYY	CHAR(8)	MDYY	‘08312012’	DATECVT(CH_MDYY, 'A8MDYY', 'MDYY')[a]	Join CH_MDYY to DC_MDYY_CHAR in DATE_CONV table	DATE(SUBSTRING(CH_MDYY,5,4) || '-' || SUBSTRING(CH_MDYY,1,2)|| '-' || SUBSTRING(CH_MDYY,3,2))
CH_YYMD	CHAR(8)	YYMD	‘20120831’	DATECVT(CH_YYMD, 'A8YYMD', 'MDYY')[a]	Join CH_YYMD to DC_YYMD_CHAR in DATE_CONV table	DATE(SUBSTRING(CH_YYMD,1,4) || '-' || SUBSTRING(CH_YYMD,5,2)|| '-' || SUBSTRING(CH_YYMD,7,2))
ZN_MDYY	ZONED (8,0)	MDYY	08312012	DATECVT(ZN_MDYY, 'I8MDYY', 'MDYY')[a]	Join ZN_MDYY to DC_MDYY_ZONED in DATE_CONV table	DATE(SUBSTRING(DIGITS(ZN_MDYY) ,5,4) || '-' || SUBSTRING(DIGITS(ZN_MDYY),1,2) || '-' || SUBSTRING(DIGITS(ZN_MDYY),3,2))

Example legacy date field name	Legacy data type	Legacy date format	Legacy example (August 31, 2012)	DB2 Web Query BIF Expression	Date dimension table	SQL function
ZN_YYMD	ZONED (8,0)	YYMD	20120831	DATECVT(ZN_YYMD, 'I8YYMD', 'MDYY')[a]	Join ZN_YYMD to DC_YYMD_ZONED in DATE_CONV table	DATE(SUBSTRING(DIGITS(ZN_YYMD),1,4) || '-' || SUBSTRING(DIGITS(ZN_YYMD),5,2) || '-' || SUBSTRING(DIGITS(ZN_YYMD),7,2))
PK_MDYY	PACKED (8,0)	MDYY	08312012	DATECVT(PK_MDYY, 'P8MDYY', 'MDYY')[a]	Join PK_MDYY to DC_MDYY_DEC in DATE_CONV table	DATE(SUBSTRING(DIGITS(PK_MDYY),5,4) || '-' || SUBSTRING(DIGITS(PK_MDYY),1,2) || '-' || SUBSTRING(DIGITS(PK_MDYY),3,2))
PK_YYMD	PACKED (8,0)	YYMD	20120831	DATECVT(PK_YYMD, 'P8YYMD', 'MDYY')[a]	Join PK_YYMD to DC_YYMD_DEC in DATE_CONV table	DATE(SUBSTRING(DIGITS(pk_yymd),1,4) || '-' || SUBSTRING(DIGITS(pk_yymd),5,2) || '-' || SUBSTRING(DIGITS(pk_yymd),7,2))
ZN_JUL5	ZONED (5,0)	YDDD Julian Date	12243	GREGDT(ZN_JUL5, 'I6YMD')	Join ZN_JUL5 to DC_JULIAN in DATE_CONV table	DATE(CASE WHEN INTEGER(SUBSTRING(DIGITS(ZN_JUL5),1,2)) > 40 THEN '19' ELSE '20' END || SUBSTR(CHAR(ZN_JUL5),1,5)) Note: Year threshold in this example is set to 40
ZN_JUL7	ZONED (7,0)	YYDDD Julian Date	2012243	GREGDT(ZN_JUL7, 'I8YYMD')	Join ZN_JUL to DC_JULIAN in DATE_CONV table	DATE(SUBSTR(CHAR(ZN_JUL),1,7))
CH_MDY	CHAR(6)	MDY	‘083112’	DATECVT(CH_MDY, 'A6MDY', 'MDYY')	Join CH_MDY to DC_MDY_CHAR in DATE_CONV table	DATE(CASE WHEN INTEGER(SUBSTRING(CH_MDY,5,2)) > 40 THEN '19' ELSE '20' END || SUBSTRING(CH_MDY,5,2) || '-' || SUBSTRING(CH_MDY,1,2)|| '-' || SUBSTRING(CH_MDY,3,2)) Note: Year threshold in this example is set to 40.
CH_YMD	CHAR(6)	YMD	‘120831’	DATECVT(CH_YMD, 'A6YMD', 'MDYY')	Join CH_YMD to DC_YMD_CHAR in DATE_CONV table	DATE(CASE WHEN INTEGER(SUBSTRING(CH_YMD,1,2)) > 40 THEN '19' ELSE '20' END || SUBSTRING(CH_YMD,1,2) || '-' || SUBSTRING(CH_YMD,3,2)|| '-' || SUBSTRING(CH_YMD,5,2)) Note: Year threshold in this example is set to 40.

<table>
<tr><th>Example legacy date field name</th><th>Legacy data type</th><th>Legacy date format</th><th>Legacy example (August 31, 2012)</th><th>DB2 Web Query BIF Expression</th><th>Date dimension table</th><th>SQL function</th></tr>
<tr><td>ZN_MDY</td><td>ZONED (6,0)</td><td>MDY</td><td>083112</td><td>DATECVT(ZN_MDY,
'P6MDY', 'MDYY')</td><td>Join ZN_MDY to
DC_MDY_ZONED in
DATE_CONV table</td><td>DATE(CASE
WHEN
INTEGER(SUBSTRING(DIGITS(ZN_MD
Y),5,2)) > 40
THEN '19'
ELSE '20'
END
||
SUBSTRING(DIGITS(ZN_MDY),5,2)
|| '-' ||
SUBSTRING(DIGITS(ZN_MDY),1,2)|
| '-' ||
SUBSTRING(DIGITS(ZN_MDY),3,2))
Note: Year threshold in this example is set to 40.</td></tr>
<tr><td>ZN_YMD</td><td>ZONED (6,0)</td><td>YMD</td><td>120831</td><td>DATECVT(ZN_MDY,
'P6YMD', 'MDYY')</td><td>Join ZN_YMD to
DC_YMD_ZONED in
DATE_CONV table</td><td>DATE(CASE
WHEN
INTEGER(SUBSTRING(DIGITS(ZN_YM
D),5,2)) > threshold THEN '19'
ELSE '20'
END
||
SUBSTRING(DIGITS(ZN_YMD),5,2)
|| '-' ||
SUBSTRING(DIGITS(ZN_YMD),1,2)|
| '-' ||
SUBSTRING(DIGITS(ZN_YMD),3,2))
Note: Year threshold in this example is set to 40.</td></tr>
<tr><td>PK_MDY</td><td>PACKED (6,0)</td><td>MDY</td><td>083112</td><td>DATECVT(PK_MDY,
'P6MDY', 'MDYY')</td><td>Join PK_MDY to
DC_MDY_DEC in
DATE_CONV table</td><td>DATE(CASE
WHEN
INTEGER(SUBSTRING(DIGITS(PK_MD
Y),5,2)) > 40
THEN '19'
ELSE '20'
END
||
SUBSTRING(DIGITS(PK_MDY),5,2)
|| '-' ||
SUBSTRING(DIGITS(PK_MDY),1,2)|
| '-' ||
SUBSTRING(DIGITS(PK_MDY),3,2))
Note: Year threshold in this example is set to 40.</td></tr>
<tr><td>PK_YMD</td><td>ZONED (6,0)</td><td>YMD</td><td>120831</td><td>DATECVT(PK_YMD,
'P6YMD', 'MDYY')</td><td>Join PK_YMD to
DC_YMD_DEC in
DATE_CONV table</td><td>DATE(CASE
WHEN
INTEGER(SUBSTRING(DIGITS(PK_YM
D),5,2)) > threshold THEN '19'
ELSE '20'
END
||
SUBSTRING(DIGITS(PK_YMD),5,2)
|| '-' ||
SUBSTRING(DIGITS(PK_YMD),1,2)|
| '-' ||
SUBSTRING(DIGITS(PK_YMD),3,2))
Note: Year threshold in this example is set to 40.</td></tr>
<tr><td>ZN_CEN
ZN_YEAR
ZN_MONTH
ZN_DAY

(Each component of the date is stored in a separate zoned decimal field.)</td><td>ZONED (2,0)
ZONED (2,0)
ZONED (2,0)
ZONED (2,0)</td><td>CC
YY
MM
DD</td><td>20
12
08
31</td><td>DATECVT(((ZN_CEN * 1000000) + (ZN_YEAR * 10000) + (ZN_MONTH * 100) + ZN_DAY), 'I8YYMD', 'YYMD')</td><td>Join legacy date fields to the following columns in DATE_CONV table:
DC_CC
DC_YY
DC_MM
DC_DD</td><td>DATE(DIGITS(ZN_CEN) ||
DIGITS(ZN_YEAR) || '-' ||
DIGITS(ZN_MONTH)|| '-' ||
DIGITS(ZN_DAY))</td></tr>
</table>

<table>
<tr><th>Example legacy date field name</th><th>Legacy data type</th><th>Legacy date format</th><th>Legacy example (August 31, 2012)</th><th>DB2 Web Query BIF Expression</th><th>Date dimension table</th><th>SQL function</th></tr>
<tr><td>CH_CEN
CH_YEAR
CH_MONTH
CH_DAY
(Each component of the date is stored in a separate character field.)</td><td>CHAR(2)
CHAR(2)
CHAR(2)
CHAR(2)</td><td>CC
YY
MM
DD</td><td>20
12
08
31</td><td>DATECVT((CH_CEN || CHYEAR || CH_MONTH || CH_DAY), 'A8YYMD', 'YYMD')</td><td>Join legacy date fields to the following columns in DATE_CONV table:
DC_CC_CHAR
DC_YY_CHAR
DC_MM_CHAR
DC_DD_CHAR</td><td>DATE(CH_CEN || CH_YEAR || ‘-’ || CH_MONTH|| ‘-’ || CH_DAY)</td></tr>
</table>

a. Denotes that the DB2 Web Query for i function is translated to SQL for optimal performance

5.3 Using dates and date components

We mentioned earlier that date and its components of Year, Quarter, Month, and Day are critical to report development. These columns are extremely useful when you must aggregate and compare elements based on the date basic components, such as sales for a given year with a quarterly breakdown.

After your dates are in the true date format, you can quickly and easily derive and use these date components in your reports.

There are two steps that are involved in using date components:

Date decomposition Creating or deriving new columns that contain the Year, Quarter, Month, and Day components for each of your dates

Date formatting Setting the display format of your date or date component.

5.3.1 Date decomposition

Synonyms can easily be enhanced to include the date decomposition fields of Year, Quarter, Month, and Day. These fields then become available for report authors to use in their reports. Enhancing synoynms in this way makes it more useful, standardizes these complex calculations, and makes them immediately available to every developer who uses the metadata.

There are several ways to decompose dates:

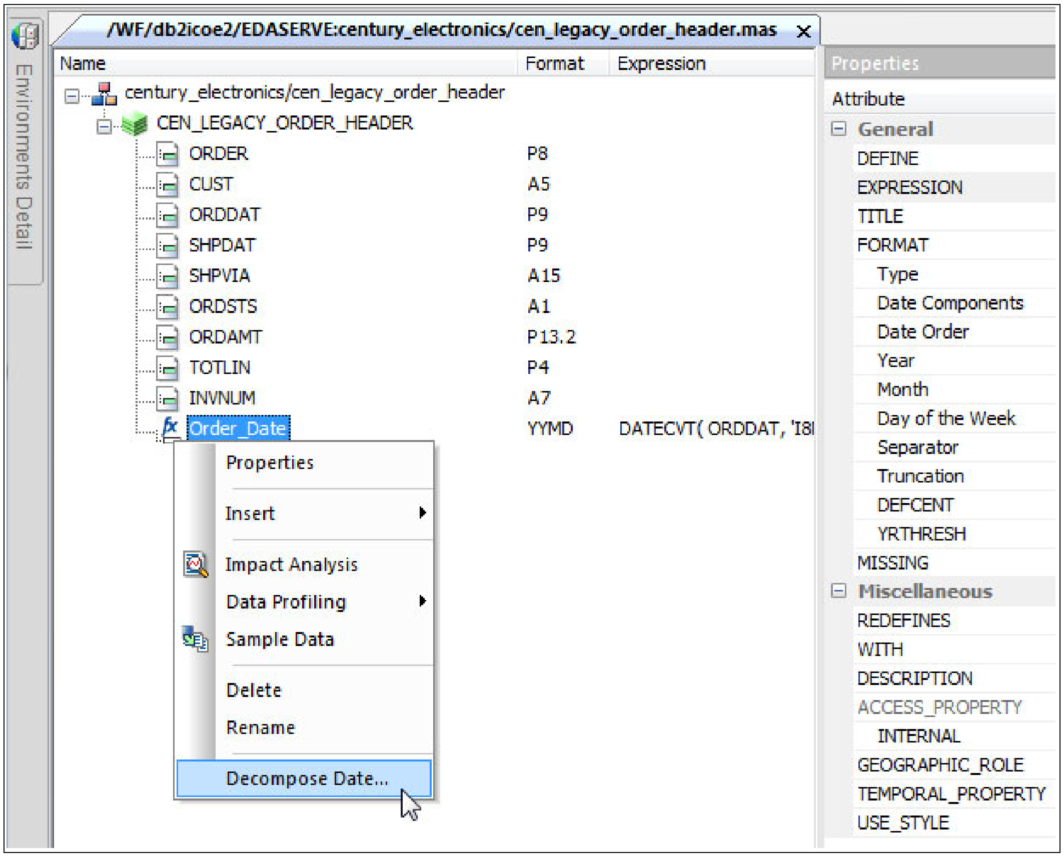
- If you used the date dimension table technique (5.2.4, “Using a date dimension table” on page 82), you automatically have Year, Quarter, Month, Day, and many more date attributes that are available in your metadata. Go to 5.3.2, “Date formatting” on page 108 to learn how to customize the formatting of these attributes.
- If you are converting dates dynamically by using BIFs (5.2.2, “Using DB2 Web Query for i functions to convert to smart dates” on page 72) or SQL functions (5.2.3, “Using SQL functions and views to convert dates” on page 78), then you can use the date decomposition feature of to decompose your dates.

DB2 Web Query for i date decomposition feature

DB2 Web Query provides a date decomposition feature within the synonym editor environment. Either the web interface or the Developer Workbench tool can be used to decompose dates.

To decompose a date field by using the Developer Workbench synonym editor, complete the following steps:

1. Open Developer Workbench, select the system on which to work, and sign in with your user ID and password.
2. Click **Data Servers** → **EDASERVE** → **Applications** → **<yourfolder>** and double-click the `.mas` file of the synonym that you want to edit. The synonym editor window opens.
3. In the synonym editor window, right-click the date field to be decomposed (`Order_Date` in this example) and select **Decompose Date**, as shown in Figure 5-32.



/WF/db2icoe2/EDASERVE:century_electronics/cen_legacy_order_header.mas

Name	Format	Expression
century_electronics/cen_legacy_order_header		
CEN_LEGACY_ORDER_HEADER		
ORDER	P8	
CUST	A5	
ORDDAT	P9	
SHPDAT	P9	
SHPVIA	A15	
ORDSTS	A1	
ORDAMT	P13.2	
TOTLIN	P4	
INVNUM	A7	
Order_Date	YYMD	DATECVT(ORDDAT, 'I8I

Figure 5-32 Decomposing a date with Developer Workbench - Step 2

4. You are presented with a window where you can select the decomposition items: Day, Month, Quarter, and Year. Click **OK**.

5.3.2 Date formatting

With any true date field, you can use the column USAGE property to change its display format to any number common formats, for example, 2016/12/31, 12/31/2016, or December 31, 2016. You can decompose your true date into its Year, Quarter, Month, and Day components and then customize their display formatting (for example, display a month value as DEC, DECEMBER, or December).

This section focuses on how to change the display formatting of date components. The display formatting of true dates falls into two main categories:

- Date formatting
- Date component formatting

In *date formatting,* the value of a date field can be assigned to an alphanumeric or integer field that uses different date display options. The reverse conversion is also possible.

In *date component formatting*, 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.

5.3.3 Changing the date format

If you want to display the dates in your reports differently from how they are stored in the database, all that is necessary is to change the **USAGE** attribute in the file's DB2 Web Query for i synonym. For example, perhaps you are storing the date columns in YYMD format and simply want those dates to appear in DMYY format on your reports.

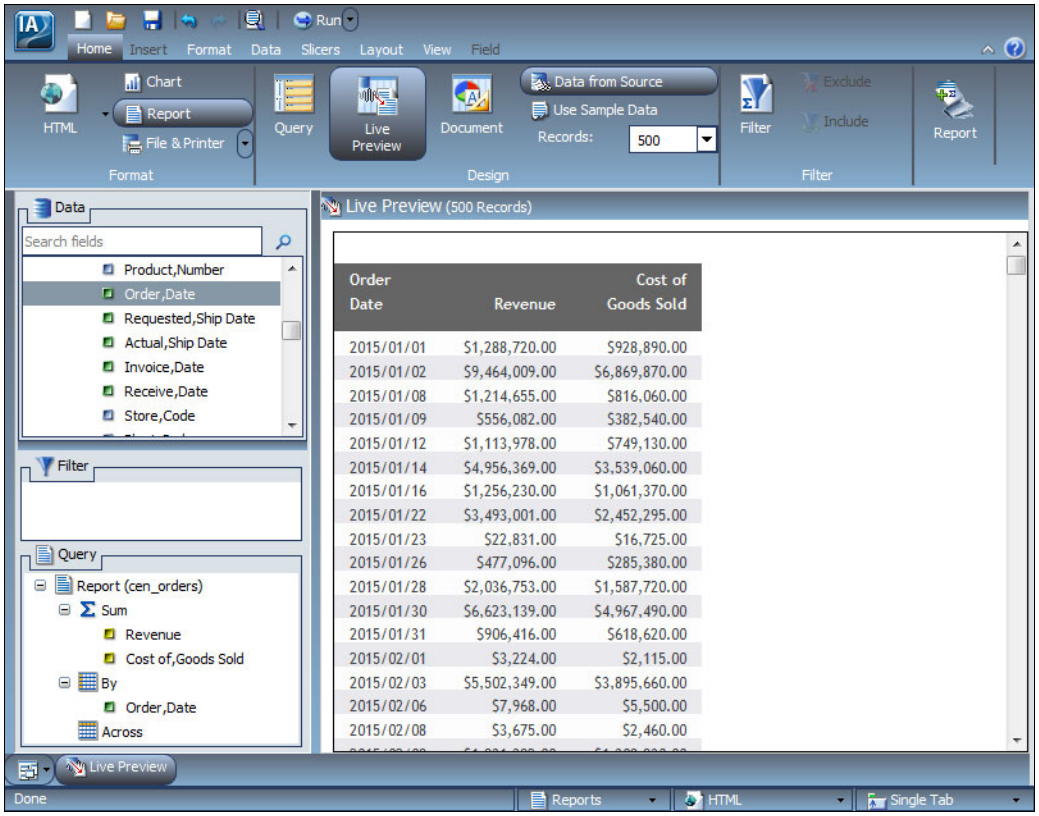
> **Note:** YYMD is the *short name* that is used to refer to the actual YYYY/MM/DD format. Similarly, MDYY refers to the full MM/DD/YYYY format.

In this exercise, you convert the order date field from the YYMD format (as it is stored in the database) so that it is displayed in MDYY format on a DB2 Web Query for i report:

1. From your browser session, log in to DB2 Web Query for i as a developer.
2. From your Century Electronics folder, create a report.
3. From the list of synonyms, select cen_orders as the report's data source.

4. In the InfoAssist tool, from list of available fields, select ORDERDATE, LINETOTAL, and COSTOFGOODSOLD and drag them into the Interactive Design View pane.

   When you finish, the report should look like the example that is provided in Figure 5-33.



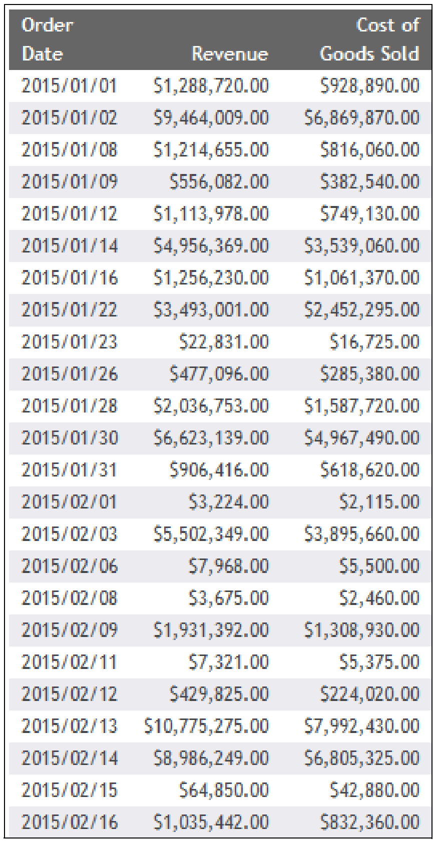
Live Preview (500 Records)

Order Date	Revenue	Cost of Goods Sold
2015/01/01	$1,288,720.00	$928,890.00
2015/01/02	$9,464,009.00	$6,869,870.00
2015/01/08	$1,214,655.00	$816,060.00
2015/01/09	$556,082.00	$382,540.00
2015/01/12	$1,113,978.00	$749,130.00
2015/01/14	$4,956,369.00	$3,539,060.00
2015/01/16	$1,256,230.00	$1,061,370.00
2015/01/22	$3,493,001.00	$2,452,295.00
2015/01/23	$22,831.00	$16,725.00
2015/01/26	$477,096.00	$285,380.00
2015/01/28	$2,036,753.00	$1,587,720.00
2015/01/30	$6,623,139.00	$4,967,490.00
2015/01/31	$906,416.00	$618,620.00
2015/02/01	$3,224.00	$2,115.00
2015/02/03	$5,502,349.00	$3,895,660.00
2015/02/06	$7,968.00	$5,500.00
2015/02/08	$3,675.00	$2,460.00

Figure 5-33 Revenue by Order Date report: YYMD format

5. Save the report as Revenue by Order Date.

6. Run the report. In this report (Figure 5-34), you see revenue and cost of goods sold ordered and grouped by order date.



Order Date	Revenue	Cost of Goods Sold
2015/01/01	$1,288,720.00	$928,890.00
2015/01/02	$9,464,009.00	$6,869,870.00
2015/01/08	$1,214,655.00	$816,060.00
2015/01/09	$556,082.00	$382,540.00
2015/01/12	$1,113,978.00	$749,130.00
2015/01/14	$4,956,369.00	$3,539,060.00
2015/01/16	$1,256,230.00	$1,061,370.00
2015/01/22	$3,493,001.00	$2,452,295.00
2015/01/23	$22,831.00	$16,725.00
2015/01/26	$477,096.00	$285,380.00
2015/01/28	$2,036,753.00	$1,587,720.00
2015/01/30	$6,623,139.00	$4,967,490.00
2015/01/31	$906,416.00	$618,620.00
2015/02/01	$3,224.00	$2,115.00
2015/02/03	$5,502,349.00	$3,895,660.00
2015/02/06	$7,968.00	$5,500.00
2015/02/08	$3,675.00	$2,460.00
2015/02/09	$1,931,392.00	$1,308,930.00
2015/02/11	$7,321.00	$5,375.00
2015/02/12	$429,825.00	$224,020.00
2015/02/13	$10,775,275.00	$7,992,430.00
2015/02/14	$8,986,249.00	$6,805,325.00
2015/02/15	$64,850.00	$42,880.00
2015/02/16	$1,035,442.00	$832,360.00

Figure 5-34 Results of Revenue by Order Date report - YYMD format

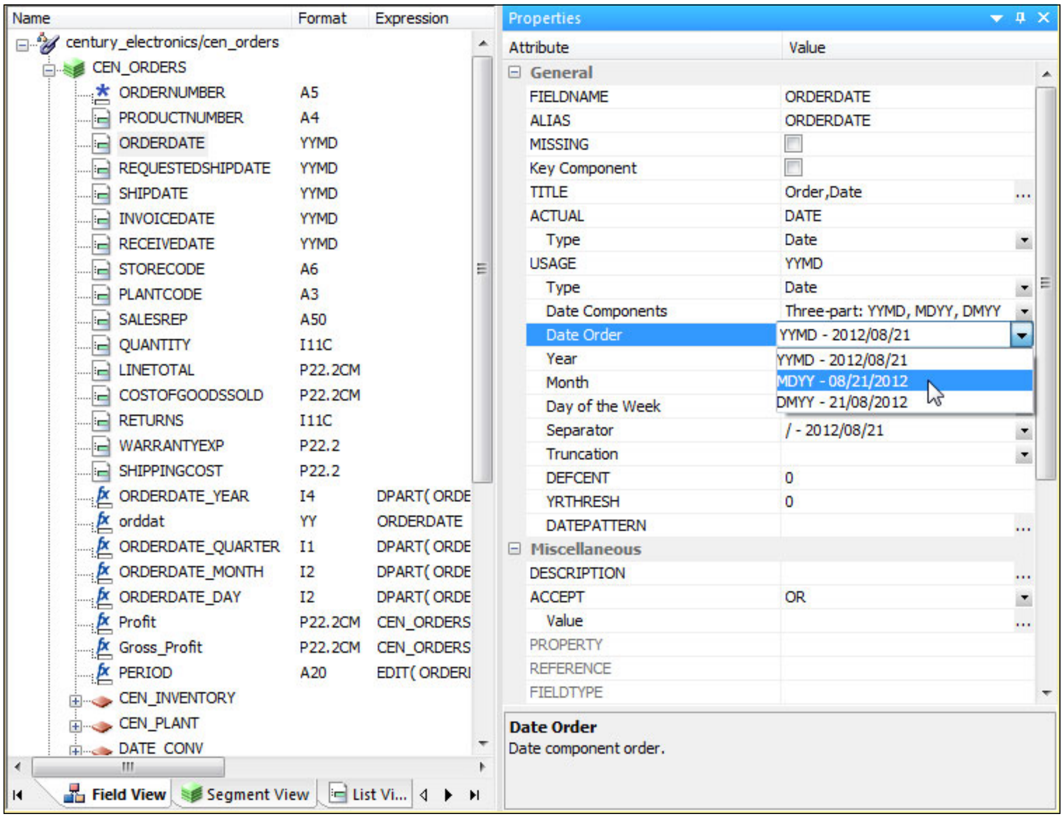
This is a useful report, but suppose that your users really want the order date shown in MDYY format. Because the order date field in your database is defined as YYMD format, you must take additional steps to override this attribute when displaying the date field in a report. These steps involve changing the usage value of the order date field as defined in the synonym of the file this report is based on.

To update the synonym use Developer Workbench.

7. Open Developer Workbench and open a connection to your environment.
8. Expand **Data Servers** → **EDASERVE** → **Applications** → **baseapp**.

9. Find and right-click the `cen_orders.mas` synonym file and select **Edit in Synonym Editor**. The synonym editor opens.

10. Select the ORDERDATE field. Notice that this is a *true* DATE field (see the ACTUAL attribute) and its USAGE value is YYMD (which is how it is defined in the database). As shown in Figure 5-35, override this attribute by selecting **MDYY** from the Date Order setting under USAGE.



Name	Format	Expression
century_electronics/cen_orders		
CEN_ORDERS		
ORDERNUMBER	A5	
PRODUCTNUMBER	A4	
ORDERDATE	YYMD	
REQUESTEDSHIPDATE	YYMD	
SHIPDATE	YYMD	
INVOICEDATE	YYMD	
RECEIVEDATE	YYMD	
STORECODE	A6	
PLANTCODE	A3	
SALESREP	A50	
QUANTITY	I11C	
LINETOTAL	P22.2CM	
COSTOFGOODSSOLD	P22.2CM	
RETURNS	I11C	
WARRANTYEXP	P22.2	
SHIPPINGCOST	P22.2	
ORDERDATE_YEAR	I4	DPART(ORDE
orddat	YY	ORDERDATE
ORDERDATE_QUARTER	I1	DPART(ORDE
ORDERDATE_MONTH	I2	DPART(ORDE
ORDERDATE_DAY	I2	DPART(ORDE
Profit	P22.2CM	CEN_ORDERS
Gross_Profit	P22.2CM	CEN_ORDERS
PERIOD	A20	EDIT(ORDERI
CEN_INVENTORY		
CEN_PLANT		
DATE_CONV		

Attribute	Value
General	
FIELDNAME	ORDERDATE
ALIAS	ORDERDATE
MISSING	□
Key Component	□
TITLE	Order,Date
ACTUAL	DATE
Type	Date
USAGE	YYMD
Type	Date
Date Components	Three-part: YYMD, MDYY, DMYY
Date Order	YYMD - 2012/08/21
Year	YYMD - 2012/08/21
Month	MDYY - 08/21/2012
Day of the Week	DMYY - 21/08/2012
Separator	/ - 2012/08/21
Truncation	
DEFCENT	0
YRTHRESH	0
DATEPATTERN	
Miscellaneous	
DESCRIPTION	
ACCEPT	OR
Value	
PROPERTY	
REFERENCE	
FIELDTYPE	

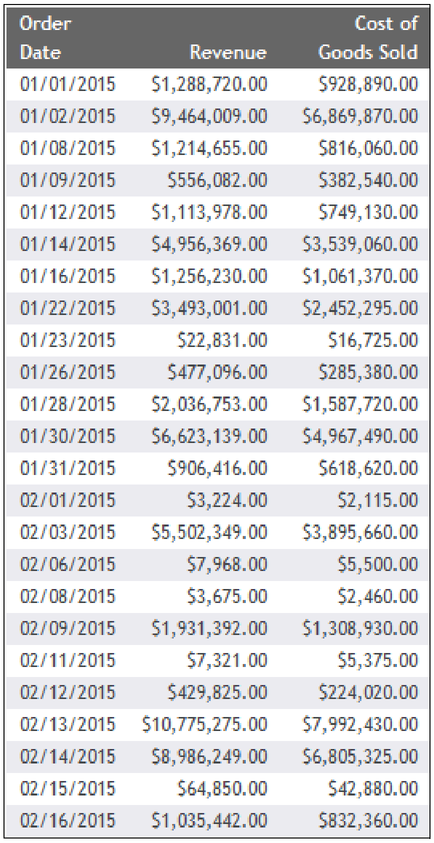
Date Order
Date component order.

Figure 5-35 Change date order to MDYY

11. The USAGE field changes to MDYY. Click the **Save** icon to save this change.

12. Return to your browser session and click the **Revenue by Order Date** report again to run it.

The Order Date column is now presented in MDYY format (Figure 5-36).



Order Date	Revenue	Cost of Goods Sold
01/01/2015	$1,288,720.00	$928,890.00
01/02/2015	$9,464,009.00	$6,869,870.00
01/08/2015	$1,214,655.00	$816,060.00
01/09/2015	$556,082.00	$382,540.00
01/12/2015	$1,113,978.00	$749,130.00
01/14/2015	$4,956,369.00	$3,539,060.00
01/16/2015	$1,256,230.00	$1,061,370.00
01/22/2015	$3,493,001.00	$2,452,295.00
01/23/2015	$22,831.00	$16,725.00
01/26/2015	$477,096.00	$285,380.00
01/28/2015	$2,036,753.00	$1,587,720.00
01/30/2015	$6,623,139.00	$4,967,490.00
01/31/2015	$906,416.00	$618,620.00
02/01/2015	$3,224.00	$2,115.00
02/03/2015	$5,502,349.00	$3,895,660.00
02/06/2015	$7,968.00	$5,500.00
02/08/2015	$3,675.00	$2,460.00
02/09/2015	$1,931,392.00	$1,308,930.00
02/11/2015	$7,321.00	$5,375.00
02/12/2015	$429,825.00	$224,020.00
02/13/2015	$10,775,275.00	$7,992,430.00
02/14/2015	$8,986,249.00	$6,805,325.00
02/15/2015	$64,850.00	$42,880.00
02/16/2015	$1,035,442.00	$832,360.00

Figure 5-36 Results of revenue by order date after changing usage to MDYY

5.4 DB2 Web Query for i date built-in functions

DB2 Web Query for i provides a rich set of built-in functions that can be used with dates. Earlier, you saw the DATECVT function in use as you converted a legacy date into a true date (smart date). This section presents a few additional more commonly used BIFs. These functions are described in more detail in the help text that is included with Developer Workbench, and in the FUNCTIONS document found off the DB2 Web Query wiki DOCUMENTATION link.

5.4.1 DPART: Extracting a date component from a date field

The `DPART` function extracts a specified component from a date field and returns it in numeric format. You might have noticed earlier that when you decompose a date in the synonym editor that the resulting date components use the `DPART` function (see the decomposed date details in Figure 5-35 on page 112).

Here is the format for `DPART`:

```
DPART (datevalue, 'component', output)
datevalue (Date) - A full component date.
```

Where:

- component (Alphanumeric): The name of the component to be retrieved, which is enclosed in single quotation marks. Here are the valid values:
  - YEAR, YY: Returns the year component.
  - MONTH, MM: Returns the month component.
  - DAY, DAY-OF-MONTH, DD: Returns the day of the month.
  - QUARTER, QQ: Returns the quarter.
- output (Integer): The field that contains the result, or the integer format of the output value, which is enclosed in single quotation marks.

Here is an example that assumes that the SHIPDATE column contains the date value 2012/09/22:

```
DPART(SHIPDATE, 'DD', 'I2')
```

A value of 22 is returned.

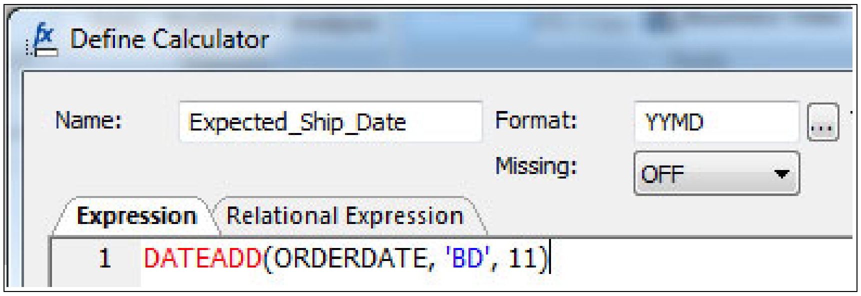
> **Note:** Using **DPART** generates an SQL statement that is submitted to the DB2 engine that performs the date arithmetic. This is important because doing this work in DB2 results in better query performance. It is for this reason that the Date Decompose feature in the Metadata Editor uses **DPART** to break up the date components.

5.4.2 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 **DATEADD** function, and the input date field must have a format such as YYMD, MDY, or JUL. Increment must be an integer. For example:

```
DATEADD(date, ‘Y/M/D/WD/BD’, increment)
```

Figure 5-37 demonstrates adding 11 business days (‘BD’) to the order date to calculate the expected ship date.



Name: Expected_Ship_Date

Format: YYMD

Missing: OFF

```
DATEADD(ORDERDATE, 'BD', 11)
```

Figure 5-37 DATEADD

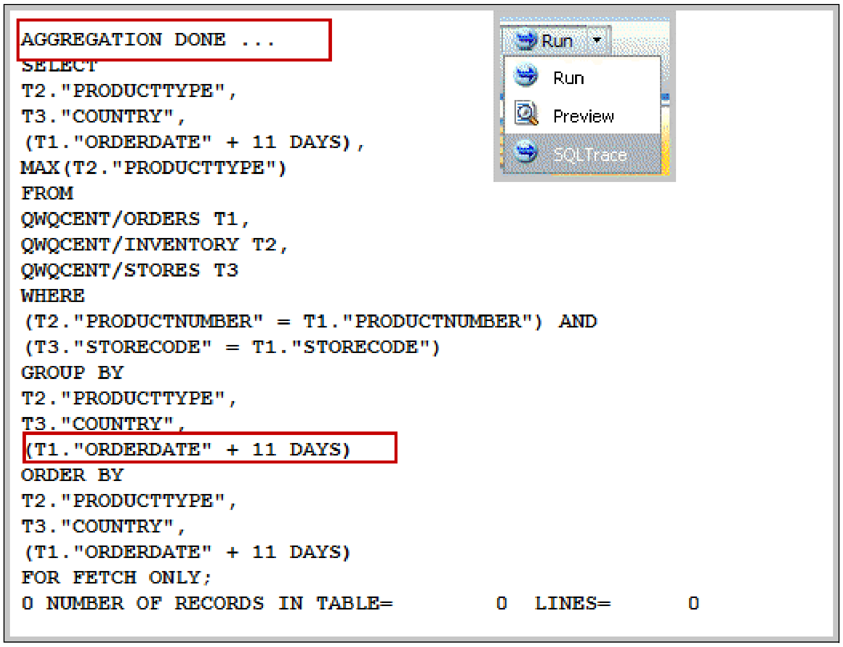
The **DATEADD** function is not required if you simply want to add months or days to a date. To add months, your result field must be in a format similar to ‘YYM’ or ‘MY’. To add days, your result field must contain days, for example, YYMD.

If you want to add 11 days to `ORDERDATE` in the previous example and do not need to worry about business or weekdays, you can replace the **DATEADD** with the following values:

```
ORDERDATE + 11
```

This creates an expected ship date of 11 calendar days in the future.

> **Note:** In many cases, `DATEADD` results in the generation of an SQL statement to submit to the DB2 engine (to carry out the date arithmetic). This is important because it results in better query performance. Some components such as ‘BD’ (Business Days) have no SQL equivalent, and thus cannot be translated. You can verify SQL translation by requesting a Run With SQL trace request within InfoAssist. A translated request looks similar to the one that is shown in Figure 5-38.

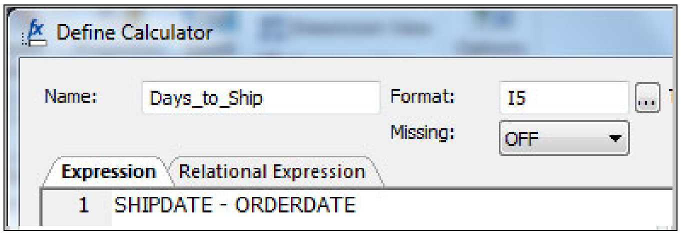


```
AGGREGATION DONE ...
SELECT
T2."PRODUCTTYPE",
T3."COUNTRY",
(T1."ORDERDATE" + 11 DAYS),
MAX(T2."PRODUCTTYPE")
FROM
QWQCENT/ORDERS T1,
QWQCENT/INVENTORY T2,
QWQCENT/STORES T3
WHERE
(T2."PRODUCTNUMBER" = T1."PRODUCTNUMBER") AND
(T3."STORECODE" = T1."STORECODE")
GROUP BY
T2."PRODUCTTYPE",
T3."COUNTRY",
(T1."ORDERDATE" + 11 DAYS)
ORDER BY
T2."PRODUCTTYPE",
T3."COUNTRY",
(T1."ORDERDATE" + 11 DAYS)
FOR FETCH ONLY;
0 NUMBER OF RECORDS IN TABLE=        0  LINES=      0
```

Figure 5-38 Translated DATEADD to SQL request

5.4.3 DATEDIF: Calculating the difference between two dates

A similar function to `DATEADD` is `DATEDIF`. You can use this function to find the difference between two dates in terms of years, months, days, weekdays, and business days. If you want to find only the difference in months or days, you do not need to use `DATEDIF`. You can create a Define field that simply subtracts two dates, as shown in Figure 5-39.



Name: Days_to_Ship
Format: I5
Missing: OFF

```
1  SHIPDATE - ORDERDATE
```

Figure 5-39 Simple subtraction of two date fields

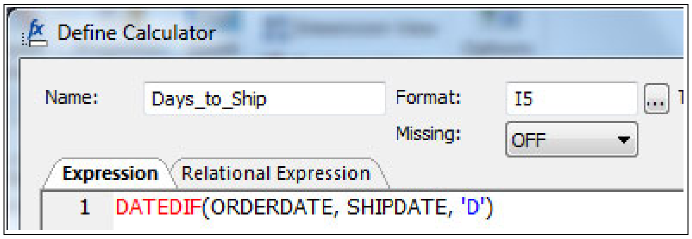
The resulting value is based on the least significant component in the order and ship dates. For example, if one of the dates is defined as YYM, then your results are in months. If both dates are defined down to the day level, then your results are in days.

The **DATEDIF** function provides the same capability but allows the developer to provide a specific unit component for the output. The format for **DATEDIF** is as follows:

```
DATEDIF('from_date', 'to_date', 'component')
```

- `from_date (Date)`: The start date from which to calculate the difference. It can be a field or a constant, for example, ‘20120101’.
- `to_date (Date)`: The end date from which to calculate the difference. Can be a field or a constant, for example, ‘20120922’.
- `component` (Alphanumeric): The unit of the value that is returned by the function. Can be one of the following (enclosed in single quotation marks):
  - Y indicates a year unit.
  - M indicates a month unit.
  - D indicates a day unit.
  - WD indicates a weekday unit.
  - BD indicates a business day unit.

When using **DATEDIF**, the format of the Define field must be defined as an I8 field, as shown in Figure 5-40.



Name: Days_to_Ship

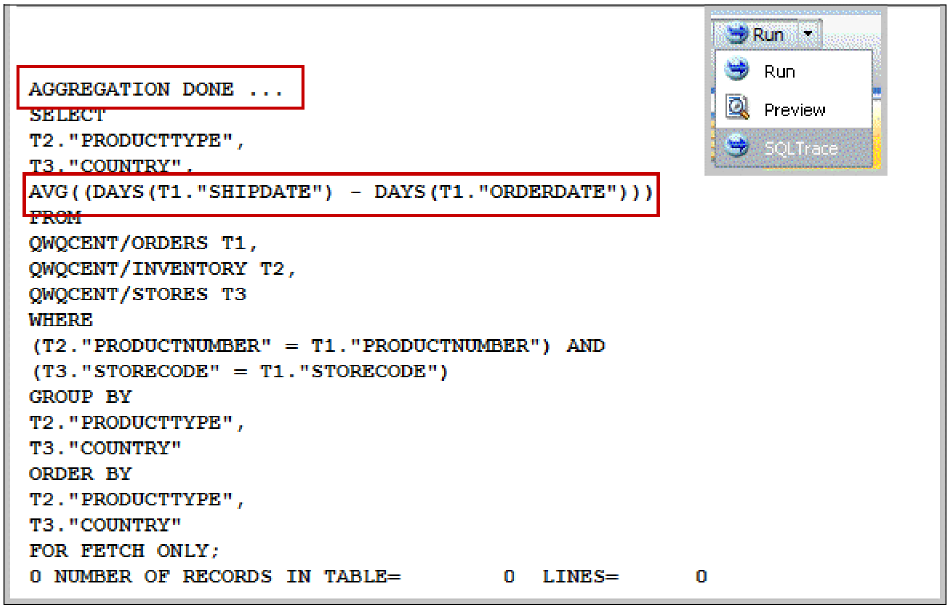
Format: I5

Missing: OFF

```
1   DATEDIF(ORDERDATE, SHIPDATE, 'D')
```

Figure 5-40 DATEDIF

Note: Once again, for query performance considerations, it is important to know that both techniques that are described in this section results in the generation of an SQL statement to submit to the DB2 engine (to carry out the date arithmetic). You can verify this by requesting a Run With SQL trace request within InfoAssist. You should see a result similar to the one that is shown in Figure 5-41.



```
AGGREGATION DONE ...
SELECT
T2."PRODUCTTYPE",
T3."COUNTRY",
AVG((DAYS(T1."SHIPDATE") - DAYS(T1."ORDERDATE")))
FROM
QWQCENT/ORDERS T1,
QWQCENT/INVENTORY T2,
QWQCENT/STORES T3
WHERE
(T2."PRODUCTNUMBER" = T1."PRODUCTNUMBER") AND
(T3."STORECODE" = T1."STORECODE")
GROUP BY
T2."PRODUCTTYPE",
T3."COUNTRY"
ORDER BY
T2."PRODUCTTYPE",
T3."COUNTRY"
FOR FETCH ONLY;
0 NUMBER OF RECORDS IN TABLE=        0  LINES=       0
```

Figure 5-41 Example of date arithmetic that is passed to the SQL and DB2 engine

5.4.4 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 5-4 lists the possible values for move-point.

```
DATEMOV(date, ‘move-point’)
```

The date field must be a full date, for example, MDYY or YYJUL.

Note: The **DATEMOV** function is *not* translated to SQL. This can result in less than optimal performance because the DB2 Web Query for i reporting server engine (and not the DB2 for i engine) processes the request.

Table 5-4 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

Move-point	Meaning
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)
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

5.5 Example: Dynamic Date Range report

In this section, you create a report that gives the user the ability to select a dynamic range of rows based on two input parameters:

- An input date
- A number of days

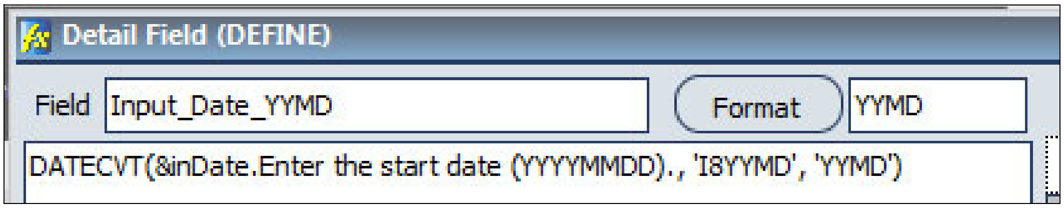
The report calculates a date range based on the two input fields. The date range is defined as follows:

- beginRange: The specified number of days before the input date
- endRange: The specified number of days after the input date

The report then returns rows in which the order date within is the specified date range. For example, if the user entered 20120915 for the date and 10 for the number of days, the report returns rows on or between September 5, 2012, and September 25, 2012.

To create this report, complete the following steps:

1. Create a report by using InfoAssist and select cen_orders at the data source.
2. Create a DEFINE field that is named `Input_Date_YYMD`. The format is YYMD and the expression is `DATECVT(&inDate.Enter the start date (YYYYMMDD)., 'I8YYMD', 'YYMD')`. An example is shown in Figure 5-42.



Field: Input_Date_YYMD

Format: YYMD

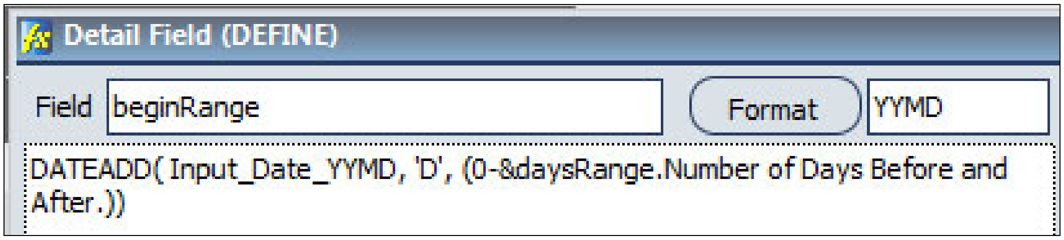
DATECVT(&inDate.Enter the start date (YYYYMMDD)., 'I8YYMD', 'YYMD')

Figure 5-42 DEFINE field for Input_Date_YYMD

The &inDate variable that is used within this Define field dynamically prompts the user for the &inDate value by using a prompt label of “Enter the start date (YYYYMMDD)”. You do not define a separate field for &inDate; it is a dynamic parameter. The DEFINE field converts the &inDate value that is entered into a smart date that is named &Input_Date_YYMD. This is an example of dynamic prompting by using the auto-prompting facility of DB2 Web Query for i.

> **Attention:** Do *not* create an input parameter for &inDate in the Selection criteria tab.

3. Create a DEFINE field named beginRange. The format is YYMD and the expression is `DATEADD( Input_Date_YYMD, 'D', (0-&daysRange.Number of Days Before and After.))`, as shown in Figure 5-43.



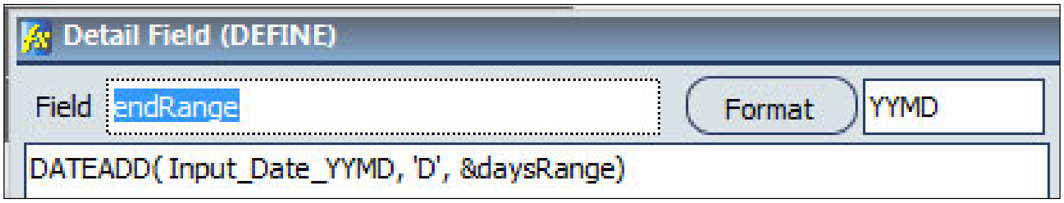
Field: beginRange
Format: YYMD

```
DATEADD( Input_Date_YYMD, 'D', (0-&daysRange.Number of Days Before and
After.))
```

Figure 5-43 DEFINE field beginRange

As in the previous example, the &daysRange variable that is used within this Define field dynamically prompts the user for the &daysRange value by using a prompt label of “Number of Days Before and After”. Again, you do *not* create a separate Define field for parameter &daysRange. The DEFINE field uses the &daysRange value that is entered to calculate a beginRange date that is the input number of days less than the &inDate input date.

4. Create a DEFINE field named endRange. The format is YYMD and the expression is `DATEADD( Input_Date_YYMD, 'D', &daysRange)`, as shown in Figure 5-44.



Field: endRange
Format: YYMD

```
DATEADD( Input_Date_YYMD, 'D', &daysRange)
```

Figure 5-44 DEFINE field endRange

This DEFINE field returns a date that is the specified number of days greater than the &inDate input date.

> **Important:** The order in which you create these DEFINE fields is important. This is because the second and third DEFINE fields contain expressions that reference the first DEFINE field. If you define them out of order, the expression fails because it cannot find and resolve DEFINE field Input_Date_YYMD.

5. Click the **Data** tab and select the **Advanced Filter** icon, as shown in Figure 5-45.

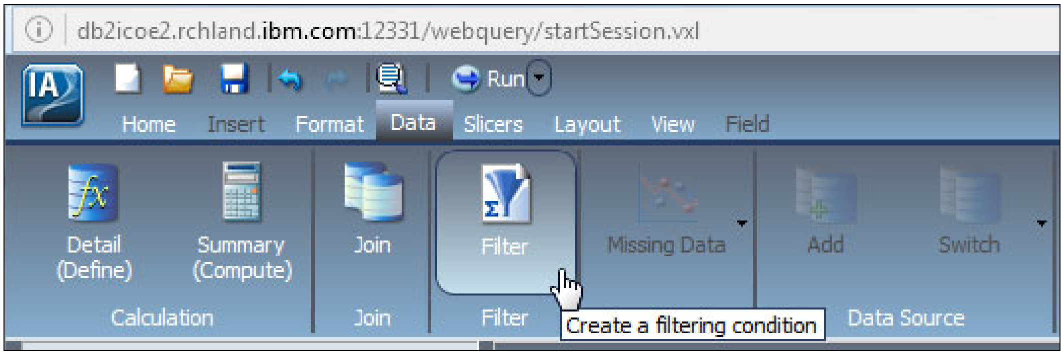
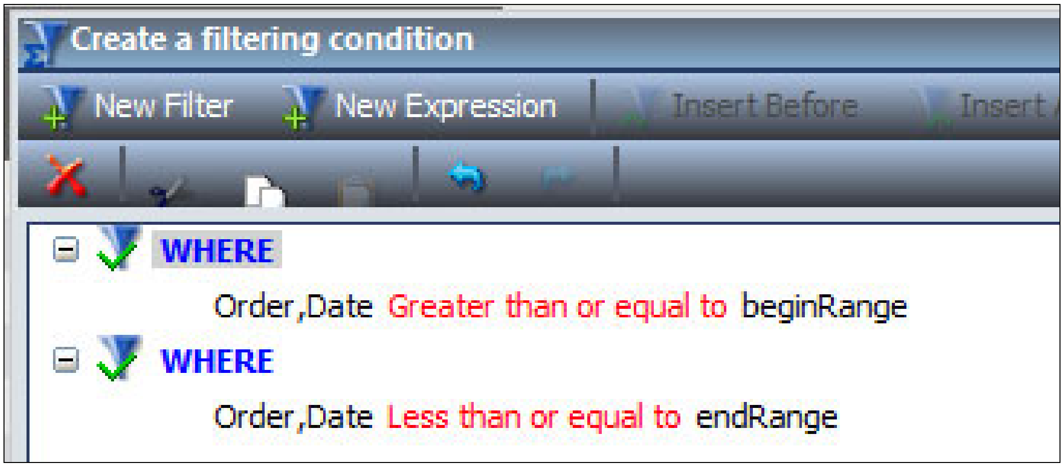


Figure 5-45 Advanced Filter icon

6. Create two filter conditions for the ORDERDATE column:
   - The first condition is GREATER THAN OR EQUAL TO the beginRange DEFINE field
   - The second condition is LESS THAN OR EQUAL TO the endRange DEFNE field.

   Select **WHERE** for both conditions. An example is shown in Figure 5-46.



WHERE
Order,Date Greater than or equal to beginRange
WHERE
Order,Date Less than or equal to endRange

Figure 5-46 Dynamic Date Range Report selection criteria

7. Return to the Field Selection tab and complete the report by selecting the following Sort-By and Sum report columns:
   - Sort-By: **ORDERDATE**, **PRODUCTTYPE**
   - Sum: **COSTOFGOODSSOLD**, **REVENUE**

An example is provided in Figure 5-47.

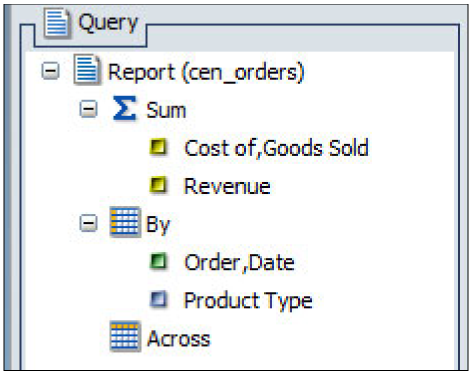
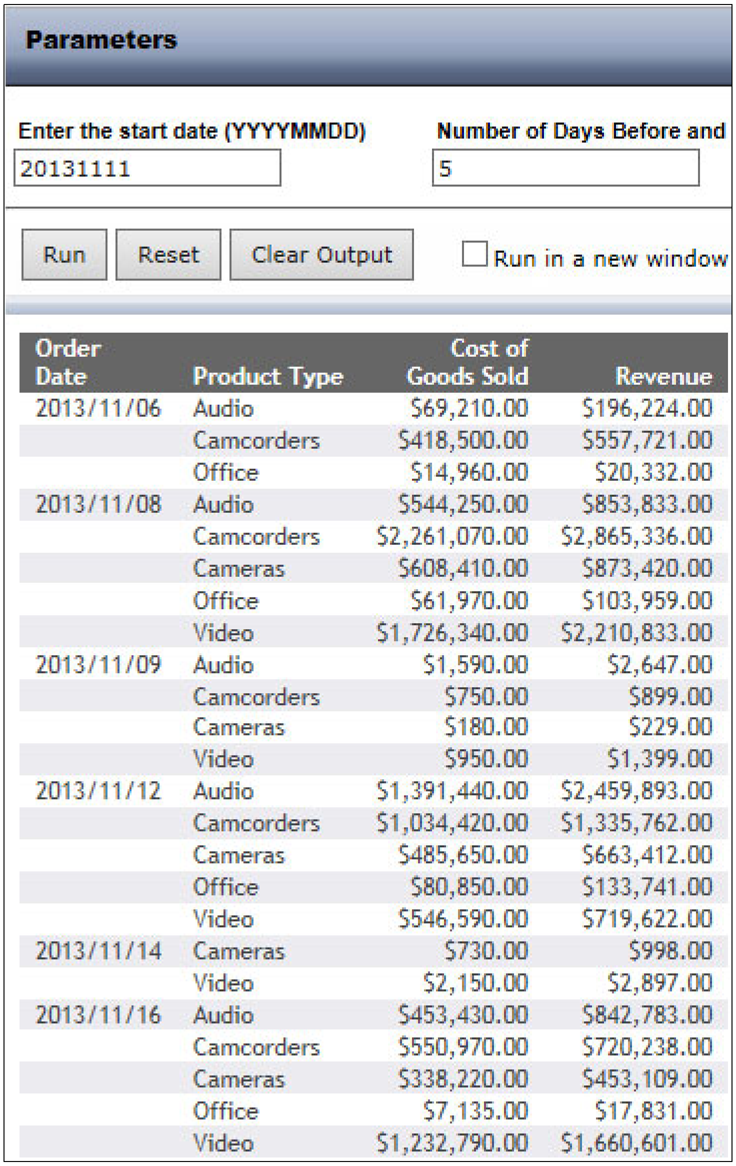


Figure 5-47 Sort-By and Sum columns

8. Save your report as Dynamic Date Range.
9. Click **Quit** to return to the DB2 Web Query for i start window.
10. Run the new Dynamic Date Range report.
11. Specify `20131111` for the inDate parameter and `5` for daysRange.

12. Click **Run**. The report returns rows for orders between 11/06/2013 and 11/16/2013, as shown in Figure 5-48.



Parameters

Enter the start date (YYYYMMDD): 20131111

Number of Days Before and: 5

□ Run in a new window

Order Date	Product Type	Cost of Goods Sold	Revenue
2013/11/06	Audio	$69,210.00	$196,224.00
	Camcorders	$418,500.00	$557,721.00
	Office	$14,960.00	$20,332.00
2013/11/08	Audio	$544,250.00	$853,833.00
	Camcorders	$2,261,070.00	$2,865,336.00
	Cameras	$608,410.00	$873,420.00
	Office	$61,970.00	$103,959.00
	Video	$1,726,340.00	$2,210,833.00
2013/11/09	Audio	$1,590.00	$2,647.00
	Camcorders	$750.00	$899.00
	Cameras	$180.00	$229.00
	Video	$950.00	$1,399.00
2013/11/12	Audio	$1,391,440.00	$2,459,893.00
	Camcorders	$1,034,420.00	$1,335,762.00
	Cameras	$485,650.00	$663,412.00
	Office	$80,850.00	$133,741.00
	Video	$546,590.00	$719,622.00
2013/11/14	Cameras	$730.00	$998.00
	Video	$2,150.00	$2,897.00
2013/11/16	Audio	$453,430.00	$842,783.00
	Camcorders	$550,970.00	$720,238.00
	Cameras	$338,220.00	$453,109.00
	Office	$7,135.00	$17,831.00
	Video	$1,232,790.00	$1,660,601.00

Figure 5-48 Dynamic Date Range report results

5.6 Date and time system variables

Table 5-5 lists the date and time variables that are available in DB2 Web Query. These variables can be specified in the headings and footers at the report and page level, and used in Define fields.

Table 5-5 Date and time variables

System variable	Description	Format or value	Example (for October 07, 2012)
`&DATE`	Returns the current date.	MM/DD/YY	10/07/2012
`&DMY`	Returns the current date.	DDMMYY	071012
`&DMYY`	Returns the current (four-digit year) date.	DDMMCCYY	07102012
`&MDY`	Returns the current date. Useful for numerical comparisons.	MMDDYY	100712
`&MDYY`	Returns the current (four-digit year) date.	MMDDCCYY	10072012
`&TOD`	Returns the current time that the query was executed.	HH.MM.SS	15.50.07
`&YMD`	Returns the current date.	YYMMDD	121007
`&YYMD`	Returns the current (four-digit year) date.	CCYYMMDD	20121007
`&DATEWtr`	Returns the full name of the day of the week.	Name of day of week	Wednesday
`&DATEMtrDYY`	Returns the name of the month followed by the day and the four-digit year.	Name of month DD, YYYY	October 7, 2012

`&DATEWtr,`
`&DATEMtrDYY`
	Returns the full name of the day of the week, followed by the name of the month, followed by the day and the four-digit year	Name of day of week, Name of month DD, YYYY	Wednesday, October 7, 2012

5.7 Date format

You can use the various date formats 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.

5.7.1 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 5-6.

Note: Using these format options might result in queries that do not translate the date formatting to SQL, which can result in a query that does not perform optimally. If you experience this, you should explore other techniques to convert your date formats. These alternative methods are described later in 5.2.7, “Converting other common legacy date formats” on page 104.

Table 5-6 Date format options

Display option	Meaning	Effect
D	Day	Displays a value 1 - 31 for the day.
M	Month	Displays a value 1 - 12 for the month.
Y	Year	Displays a 2-digit year.
YY	Four-digit year	Displays a 4-digit year.
T	Translate month	When used with M in a date (MT or TM), the 3-letter abbreviation for the month in uppercase is displayed.
t	Translate month	When used with M in a date (Mt or tM), the 3-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 3-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 4-digit year and the number of days counting from January 1. For example, January 3, 2001 in Julian format is 2001003.

Table 5-7 shows samples of output for various date formatting options.

Table 5-7 Sample output for date formatting options

Translation	Display
MT	JAN
Mt	Jan
MTR	JANUARY
Mtr	January
WR	MONDAY
wr	Monday
Q	Q1
YQ	07Q1

5.7.2 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, 12 Q3 or Q3 2012). 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 5-8 shows the `FORMAT` specifications that can be used to change the separating character.

Table 5-8 Date separators

Format	Display
YMD	12/09/22
Y.M.D	12.09.22
Y-M	12-09
YBMBD	12 09 22 (The letter B signifies blank spaces.)
Y|M|D	120922 (The concatenation symbol | eliminates the separation character.)

5.7.3 Using date fields

Table 5-9 shows valid examples of specifying dates.

Table 5-9 Examples of specifying the dates

Situation	Natural date literal
In WHERE screening	WHERE MYDATE IS 'SEP 22 2012'
In arithmetic expressions	MYDATE - '2012 SEP 22'
In computational date comparisons	IF MYDATE GT '22 SEP 2012'

5.7.4 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 that have elapsed since January 1, 1999:

```
YEARTODATE = ORDERDATE - 'JAN 1 1999' ;
```

Note: Changing the usage field might be enough if your date fields are defined as true date fields in your database and all you want to do is change the month-day-year order of how the date is displayed. However, as mentioned earlier, if you are like many IBM i shops, you probably have date fields that are defined as a data type other than date or time stamp (because these data types were not supported by the RPG compiler until V3R1 of the operating system). Therefore, you might have date fields that are defined as packed decimal (8,0), which contain the value 04102008 to represent the date April 10, 2008.

6

IBM DB2 Web Query for i Security Center

User management is handled through the IBM DB2 Web Query for i Security Center. This is a GUI that simplifies user administration because it is tightly integrated with License Manager and dynamically updates license information.

All users that log into DB2 Web Query must either be specifically registered through the Security Center (or REGWQUSR CL command) or, under Standard Edition, belong to a group profile which itself is registered through the Security Center.

Once a user is registered, they can subsequently be assigned to one or more roles, either as a global group or via folder groups. For example, a user can be assigned the role of an administrator. Their role determines what they can do within the product. In effect, roles provide an additional level of security and separation of duties.

6.1 Security architecture and concepts

The security architecture consists of a new security model that is called *Universal Object Access (UOA) Security*. UOA Security provides more granular control of objects. Rules define what a user can or cannot do.

6.1.1 Security architecture

UOA is composed of the following elements:

Users Unique ID.

Folders Contains Managed Reporting (MR) content. There are no limits to the folder depth.

Groups (Roles) Container of users or subgroups with similar capabilities. Each top-level folder has six groups/roles: Folder-Run, Folder-Analyst, Folder-Developer, Folder-DBA, Folder-Sched, and Folder-Admin.

Permission Sets Group of permitted or denied operations. These are predefined for the Folder-xxx groups.

6.1.2 Security concepts

This section outlines role-based security concepts. Groups are predefined to represent a specific set of functions or roles. Global groups define a role at the product level and apply across folders, and folder groups define a role at the top-level folder level.

There are two global groups that are defined, as shown in Table 6-1.

Table 6-1 Global groups

Group name	Role description
WebQueryAdministrator	Can perform all functions in DB2 Web Query for i and can access all folders.
DevWorkBench	Can connect to DB2 Web Query for i by using Developer Workbench.

The six folder groups (roles) are defined, as shown in Table 6-2.

Table 6-2 Groups that are associated to a top-level folder

Group name	Role Description
Folder-run	Can run reports in the respective folder.
Folder-analyst	Can develop, run, and schedule *private* reports.
Folder-dev	Can develop and run both *private and managed (published)* reports.
Folder-dba	Can only manage metadata.
Folder-sched	Can schedule reports to run through Report Broker.
Folder-admin	Can manage users in the respective folder. A folder-admin group cannot add or release (delete) a developer or group license.

6.2 IBM i security

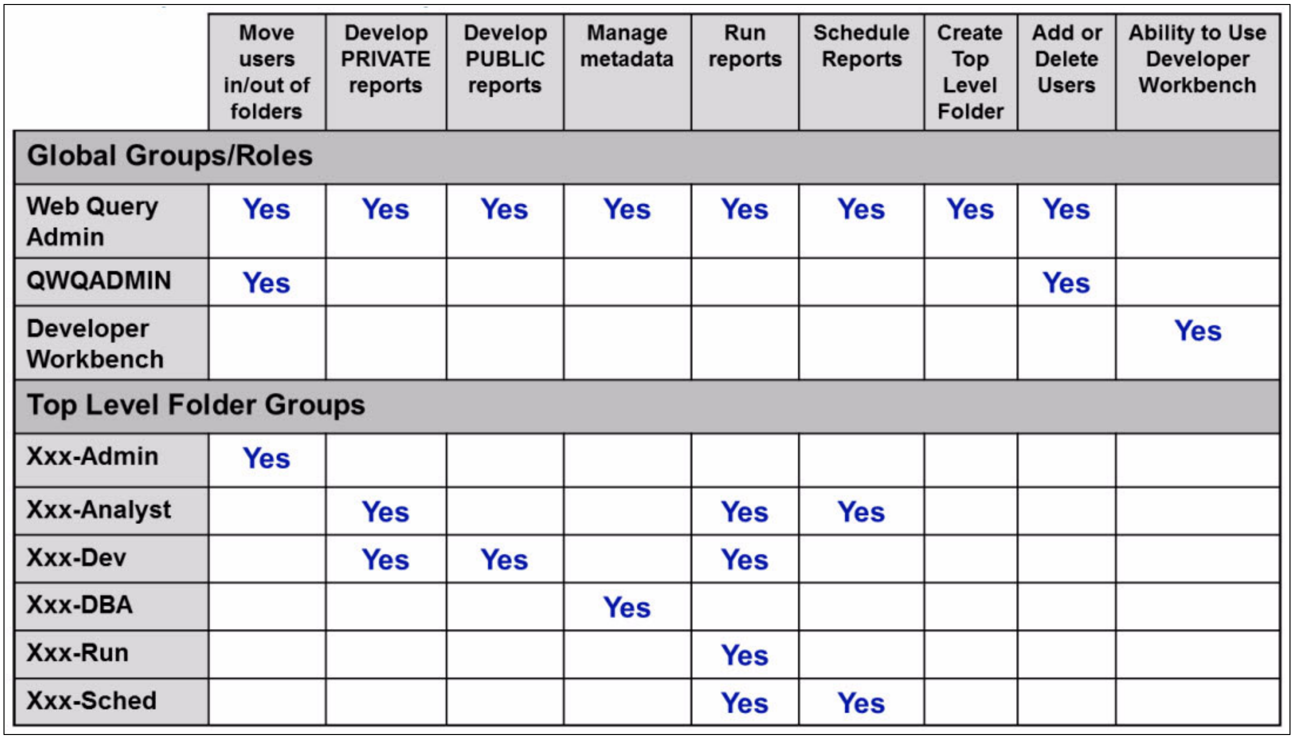
Regardless of roles that are granted, users still need IBM i OS object-level authority to all objects that are used in a report, which includes the following objects:

- Tables/physical files
- SQL views
- Stored procedures
- IFS files
- Query/400 definitions

Note: A user could have access to a report, but it does not mean that the user can run it and get returned data if they are not authorized to the underlying object.

6.3 Groups and their capabilities

Section 6.1.2, “Security concepts” on page 128 showed which groups are generated when you create a top- level folder. Figure 6-1 shows the global and the top-level folder groups and their capabilities.



	Move users in/out of folders	Develop PRIVATE reports	Develop PUBLIC reports	Manage metadata	Run reports	Schedule Reports	Create Top Level Folder	Add or Delete Users	Ability to Use Developer Workbench
Global Groups/Roles									
Web Query Admin	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	
QWQADMIN	Yes							Yes	
Developer Workbench									Yes
Top Level Folder Groups									
Xxx-Admin	Yes								
Xxx-Analyst		Yes			Yes	Yes			
Xxx-Dev		Yes	Yes		Yes				
Xxx-DBA				Yes					
Xxx-Run					Yes				
Xxx-Sched					Yes	Yes			

Figure 6-1 Groups and their capabilities

6.4 Top-level folders and subfolders

This section explains the top-level folders and subfolders.

6.4.1 Top-level folders

Top-level folders store reporting objects and can be used to segregate those objects. For example, you may want to have a top-level folder for Human Resources (HR) and another folder for Sales where users can be assigned to one or more folders. In our tutorials we use a top-level folder that is named Century Electronics.

A user can have a different set of roles for each folder. For example, a user can be assigned to run reports only in Sales and HR, but takes on a developer/DBA role in Century Electronics.

Each top-level folder is created with its own set of DB2 Web Query for i groups that define the authorization rules for the folder (see 6.1.2, “Security concepts” on page 128).

The Common top-level folder exists for all installations. The purpose of this folder is to contain content that can be accessed by all users. If you prefer not to display the Common top-level folder, there is an option to hide it. You can do this by right-clicking the Common top-level folder and selecting **Hide** (see Figure 6-2).

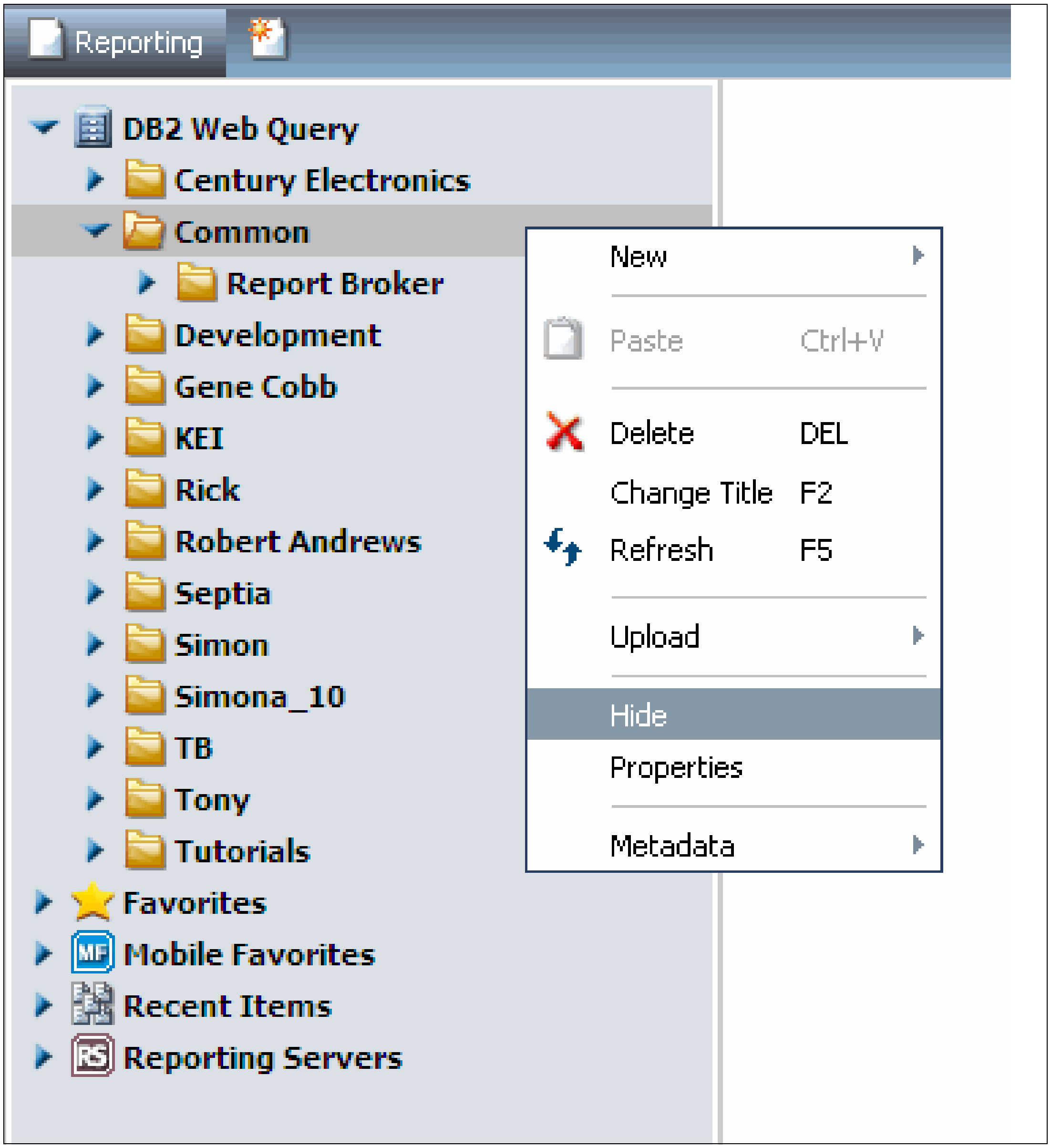


Figure 6-2 Hide the common top-level folder

Note: To be able to hide the Common folder, you must be a member of the *WebQueryAdministrator* group.

A Top Level folder consists of different reporting objects:

- Reports:
  - Reports, graphs and charts, dashboards
- Other Files
  - Uploaded images (`.gif` and `.jpg`), HTML files, and other files that are used to customize and enhance reports
- Subfolders: Allows for logical segments of reports within the top-level folder group
- Schedules and Distribution Lists used for automated job execution and distribution

When you create a top-;evel Folder, an *application folder for storing metadata (apps/folder_name)* is also created with the same name. Synonyms (the Web Query term for a metadata object) are secured in the IFS by using an authorization list that is automatically updated based on a user's authorization to the top-level folder:

- Run users require only READ.
- DBA users require only READ/WRITE.

Each top-level folder that is created automatically has the six folder-based groups that are created in the DB2 Web Query for i repository.

For example, if you add a top-level folder that is named "Century Electronics," the following six groups are automatically created:

1. Century_Electronics-run: Can run reports in the Century Electronics folder.
2. Century_Electronics-analyst: Can develop and run reports in private folders within the Century Electronics folder.
3. Century_Electronics-developer: Can develop, run, and publish reports in a published folder within the Century Electronics folder.
4. Century_Electronics-dba: Can manage metadata in the Century Electronic folder application directory.
5. Century_Electronics-sched: Can manage schedules and distribution lists in the Century Electronics folder.
6. Century_Electronics-admin. Can add or remove a user from the Century Electronics group.

> **Note:** Users are added to one or more groups to provide the functions that they require to perform their job. This is done by using the DB2 Web Query for i Security Center.

6.4.2 Subfolders

Subfolders are created within top-level folders and can go multiple levels deep. Your reporting objects (reports, charts, documents, and dashboards) are created in these subfolders. You can create reporting objects in the top-level folder (and not a subfolder) if you want. However, if you use subfolders, you can use further logical segmentation and organization of reporting objects within the top-level folder.

You may create as many levels of subfolders as you need to organize your reporting objects.

> **Note:** Top-level folder authority is applied to all subfolders and the metadata (synonyms).

6.4.3 Controlling accessibility to subfolders and reporting objects

In addition to the security features that are already described, DB2 Web Query has additional ways for developers to control if and how content in the top-level folder is accessed by other users. This task is accomplished by using two additional properties: *publication status* and the `hide/show attribute`. These properties can be applied to both subfolders and reporting objects (reports, charts, dashboards, compound documents, HTML files, and so on) within those folders.

Publication status

The publication status determines whether other users can see and run that particular element. The two values for publication status are as follows:

- Private

  Also referred to as Unpublished, this status means that only the developer that created the report or subfolder has access to that element. It does not appear on the resource tree for other users, and it cannot be run from other interfaces. This restriction gives the developer an opportunity to continue working on the report until it is ready to be promoted for others to see and run. This value is the default value. Therefore, the developer must publish any report before other users can access it.

- Published

  When a subfolder or reporting object is in Published status, all users who have Run authority to the top-level folder can see the report in the resource tree. They can also run the report from any supported interface, such as the resource tree, the Excel plug-in, or from a web services call .

Hide/show attribute

In addition, there is also a property that is known as the hide/show attribute, which can be applied to top-level folders, subfolders, and reporting objects. The two values for this property are as follows:

- Show

  The subfolder or report is seen by all users with Run authority to the top-level folder. This value is the default value.

- Hide

  The folder or report is seen only by the developer from the resource tree. However, it can still be referenced and run from other interfaces, such as a drill down from another report, or called from a web services. The developer must explicitly take this action to activate this attribute.

  You might be wondering why you might want to publish a report so that it is accessible to all users and then hide it. This technique might be appropriate if you have many drill-down reports (where you have a summary or parent report and allow your users to drill down to other reports or charts with more details). You might not want to expose the drill-down reports or charts in the resource tree (because they are meant to provide more information only through the drill-down process). If you hide these detailed reporting objects, you can prevent your users from running them directly from the resource tree. A side benefit of this is that your folders look more organized and less cluttered.

> **Tip:** Create a specific subfolder for all of your drill-down reports within a top-level folder. Then, hide the subfolder so none of that content appears on the resource tree.

Determining the publication status and hide/show attribute

Publication status and the hidden attribute are identified in the following ways. An icon is placed to the left of the element. The color and transparency of the icon inform you about both the publication status and whether the element is hidden or not. The rules are as follows:

- Unpublished: The icon is gray.
- Published: The icon has color.
- Hidden: The icon is semi-transparent.
- Not hidden: The icon is opaque.

A folder example of this visual representation is shown in Figure 6-3.

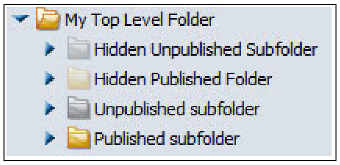

My Top Level Folder
Hidden Unpublished Subfolder
Hidden Published Folder
Unpublished subfolder
Published subfolder

Figure 6-3 Icons for publication status and hidden attribute

For reporting objects such as reports, graphs, and dashboards, the icons themselves are different but follow the same rules as described above for representing publication status and hidden attribute, as shown in Figure 6-4.

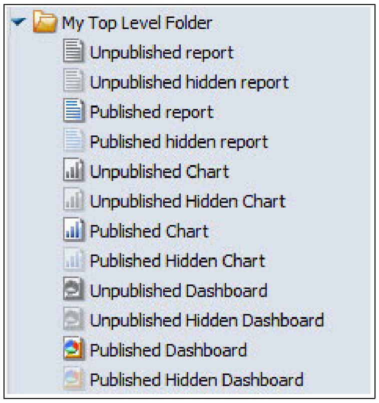
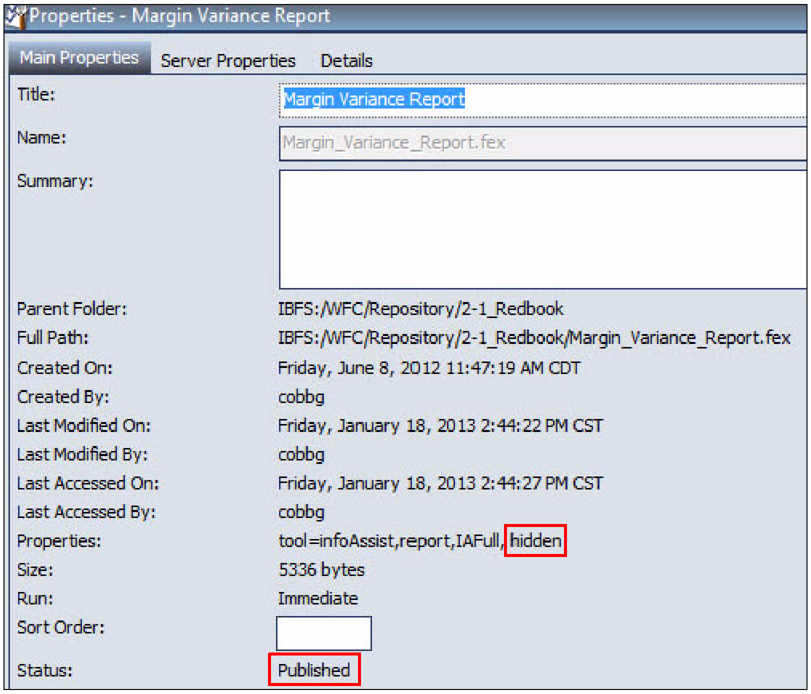

My Top Level Folder
Unpublished report
Unpublished hidden report
Published report
Published hidden report
Unpublished Chart
Unpublished Hidden Chart
Published Chart
Published Hidden Chart
Unpublished Dashboard
Unpublished Hidden Dashboard
Published Dashboard
Published Hidden Dashboard

Figure 6-4 Icons for reporting the object publication status and hidden attribute

Notice the use of color and transparency to help you understand if the content is published and whether it is hidden from the resource tree of other users.

Note: You might also notice that published elements appear in bold font. This feature was enabled in the first two group PTFs of the product, was inadvertently removed, and later added back in again.

If you find it hard to visually distinguish the differences in the icons and are unsure of the values of publication status and hide/show attribute for a particular reporting object or subfolder, you can obtain this information by right-clicking the report or folder and selecting **Properties**, as shown in Figure 6-5.



Title:	Margin Variance Report
Name:	Margin_Variance_Report.fex
Summary:	
Parent Folder:	IBFS:/WFC/Repository/2-1_Redbook
Full Path:	IBFS:/WFC/Repository/2-1_Redbook/Margin_Variance_Report.fex
Created On:	Friday, June 8, 2012 11:47:19 AM CDT
Created By:	cobbg
Last Modified On:	Friday, January 18, 2013 2:44:22 PM CST
Last Modified By:	cobbg
Last Accessed On:	Friday, January 18, 2013 2:44:27 PM CST
Last Accessed By:	cobbg
Properties:	tool=infoAssist,report,IAFull,hidden
Size:	5336 bytes
Run:	Immediate
Sort Order:	
Status:	Published

Figure 6-5 Properties of a report

In Figure 6-5, you see that the report is hidden and published.

Publishing and unpublishing content

As mentioned, new reports are private and not accessible to other users. When the report is ready for promotion, you can publish it by right-clicking the report and selecting **Publish**. An example is shown in Figure 6-6.

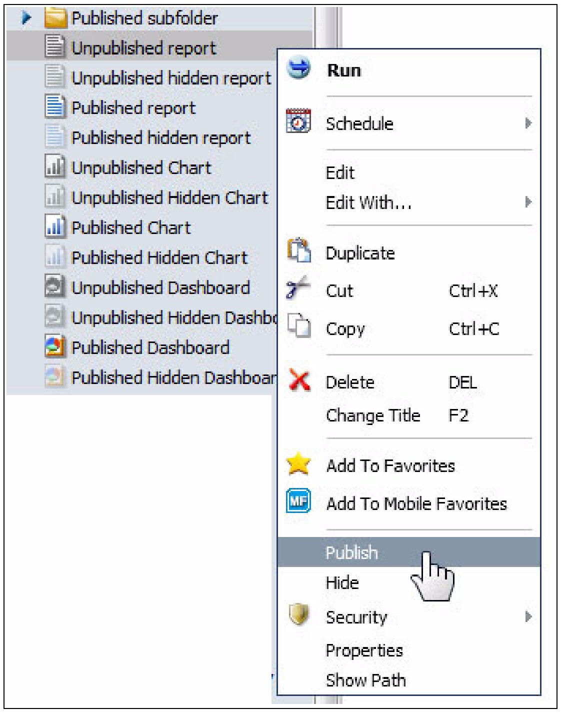


Figure 6-6 Publish a report

Published content can also be unpublished. You might want to do this to change an existing report; this option effectively takes it out of production and allows you to edit and test it. To unpublish a report, right-click it and select **Unpublish**, as shown in Figure 6-7.

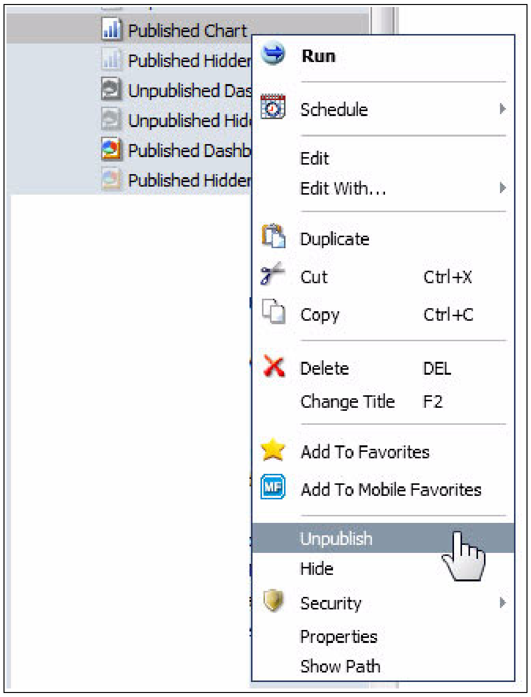


Figure 6-7 Unpublish a report

You can also publish/unpublish at the subfolder level. Simply right-click the subfolder and select the appropriate option (**Publish** or **Unpublish**). This action changes the publication status of all content in the subfolder (including nested subfolders and reporting objects within).

Tip: You cannot publish/unpublish at the top-level folder. However, you can process multiple reports at once by holding down the Ctrl key, selecting the reports, and selecting the **Publish** or **Unpublish** option.

Hiding and showing content

By default, reports are not hidden. You can change this attribute by right-clicking the report and selecting **Hide**. An example is shown in Figure 6-8.

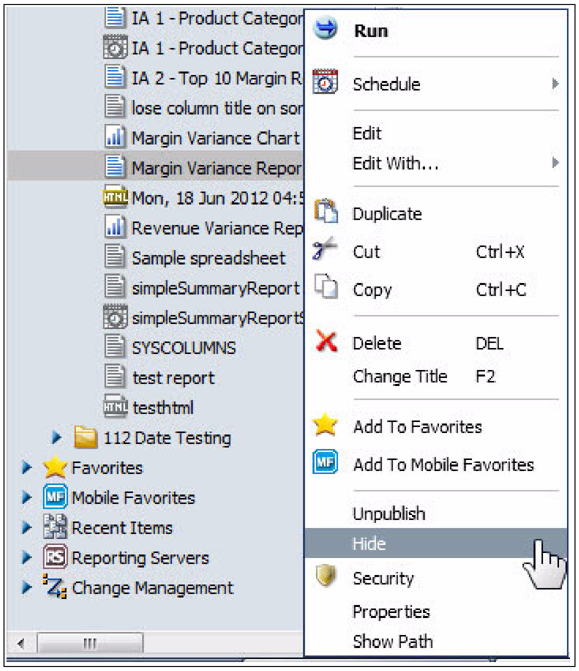


Figure 6-8 Hide report

If you hide a report and decide later that it should be displayed on the resource tree, you can accomplish this by right-clicking the hidden report and selecting **Show**. An example is shown in Figure 6-9.

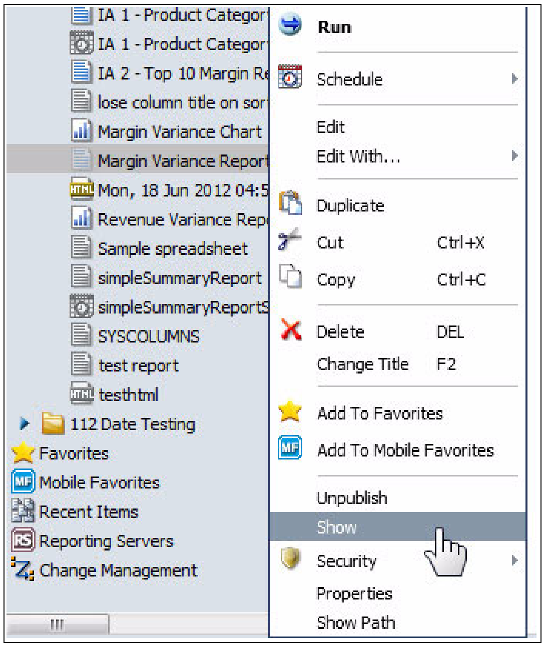
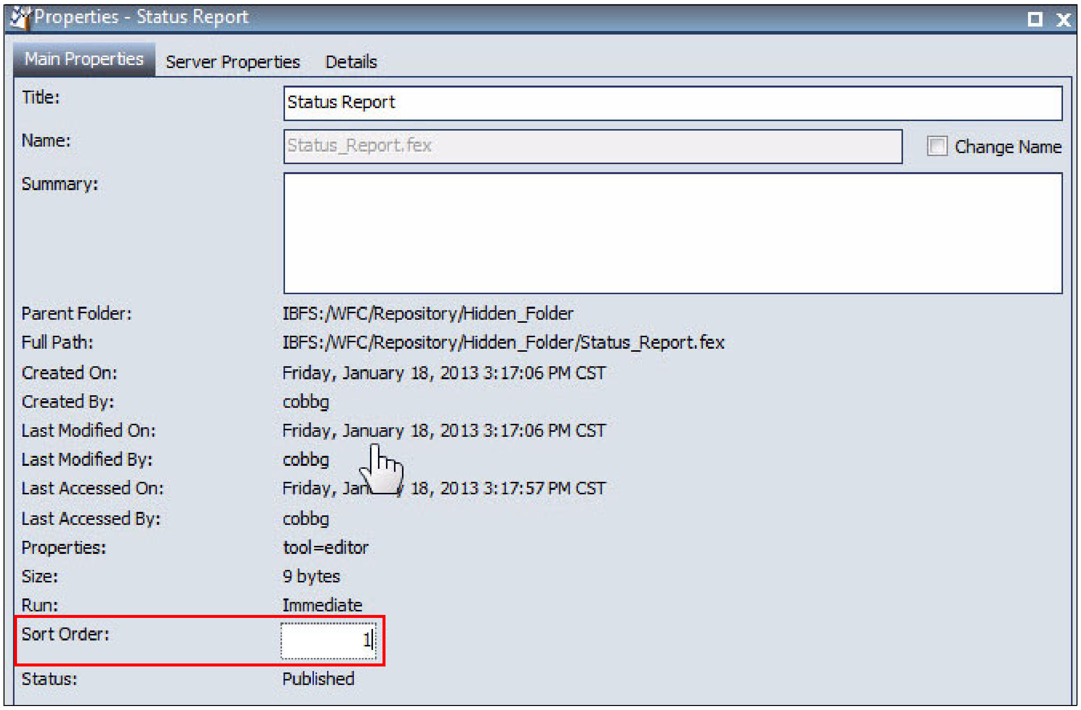


Figure 6-9 Show report

6.4.4 Controlling order of elements

By default, subfolders (and the content within those subfolders) appear in the resource tree in alphabetical order by name. In some cases, you might want to override this order and specify your own order. This task can be accomplished by using the Sort Order property. Simply right-click the report or folder, select **Properties**, and specify a value for the Sort Order setting. An example is shown in Figure 6-10.



Title:	Status Report
Name:	Status_Report.fex
Summary:	
Parent Folder:	IBFS:/WFC/Repository/Hidden_Folder
Full Path:	IBFS:/WFC/Repository/Hidden_Folder/Status_Report.fex
Created On:	Friday, January 18, 2013 3:17:06 PM CST
Created By:	cobbg
Last Modified On:	Friday, January 18, 2013 3:17:06 PM CST
Last Modified By:	cobbg
Last Accessed On:	Friday, January 18, 2013 3:17:57 PM CST
Last Accessed By:	cobbg
Properties:	tool=editor
Size:	9 bytes
Run:	Immediate
Sort Order:	1
Status:	Published

Figure 6-10 Specify sort order

The elements appear in the folder in ascending order for the Sort Order value. An example is shown in Figure 6-11.

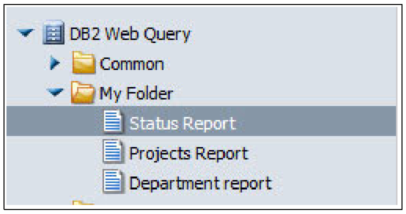


Figure 6-11 Results of specifying the sort order

Notice that the reports and charts are not in alphabetical order because each one has a Sort Order setting that is specified.

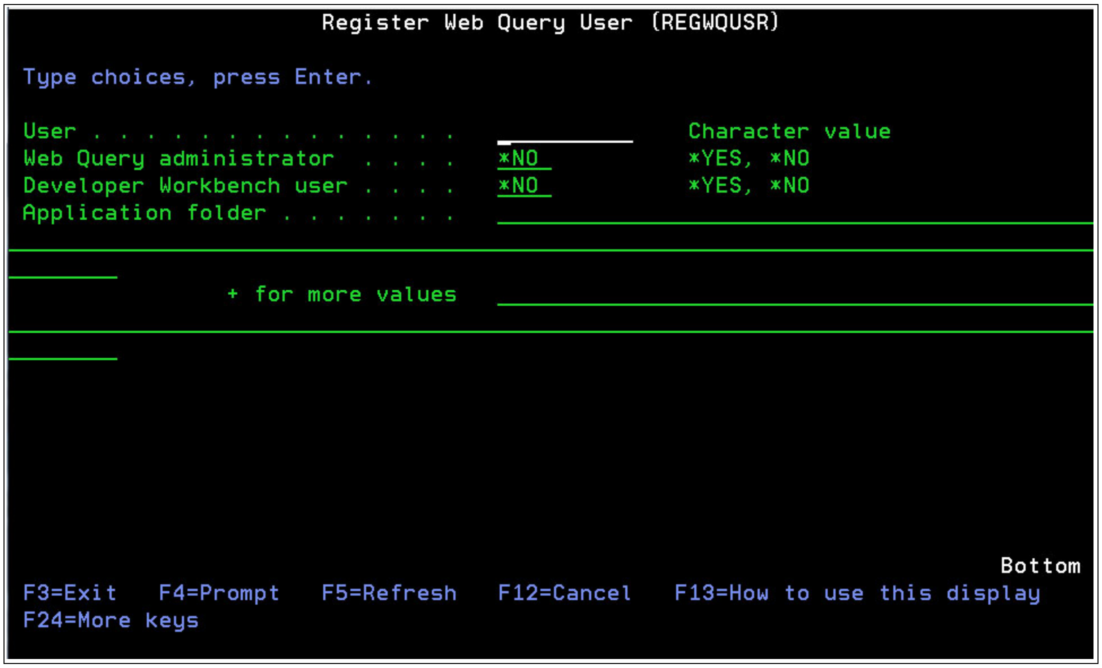
Tip: Elements that have a Sort Order value that is specified appear *after* the elements without a Sort Order value. If you want complete control over all the ordering, you must specify a Sort Order value for all of the elements in the folder.

6.5 The Register User (REGWQUSR) command

The **REGWQUSR** command provides a CL interface for adding users and assigning folder permissions to them. The command complements the Security Center, and provides a programmatic way to register users. The command allows an administrator to do the following tasks:

- Register a user profile as a licensed Developer User.
- Assign a licensed user to one of more folder groups.

The command parameters are shown in Figure 6-12.



```
                      Register Web Query User (REGWQUSR)

 Type choices, press Enter.

 User . . . . . . . . . . . . . .   __________    Character value
 Web Query administrator  . . . .   *NO           *YES, *NO
 Developer Workbench user . . . .   *NO           *YES, *NO
 Application folder . . . . . . .   

                + for more values   

                                                                    Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

Figure 6-12 The REGWQUSR command

Users who are authorized to run the **REGWQUSR** command are QWQADMIN, members of the WebQueryAdministrator group, and users with *ALLOBJ authority. You do not have to start DB2 Web Query to run the **REGWQUSR** command.

Other administrative tasks, such as revoking a user's license or removing a user from a folder, must be performed from the Security Center.

7

Accessing additional data sources

DB2 Web Query is automatically configured to access your local DB2 for i database. And accessing remote DB2 for i databases is as simple as defining a new connection.

DB2 Web Query also provides additional Adapters that are optimized to work seamlessly with several kinds of additional data sources. Although most of your data is likely in DB2 for i, access to data that is stored in other databases might be necessary when creating reports.

These adapters extend your reporting environments by incorporating access to data that is stored in other databases. They are also available to be used for extracting data to subsequently load into a DB2 for i data mart or data warehouse using the DB2 Web Query DataMigrator ETL Extension described in Chapter 8.

The adapters that are introduced are the Microsoft SQL Server, MySQL, PostgreSQL, and generic JDBC adpaters. Also covered is accessing other DB2 family members, such as DB2 for Linux, UNIX, or Windows or IBM z/OS®, and an application adapter providing pre-built metadata for Oracle JD Edwards EnterpriseOne and World applications storing their data in DB2 for i.

As of this publication, note that DB2 family access is available in Express Edition. Standard Edition is a requirement for non DB2 database access as well as the separate feature for the JDE application adapter.

7.1 Connecting to DB2 for Linux, UNIX, or Windows or z/OS

You can use DB2 Web Query to connect to DB2 family members such as Linux, UNIX, or Windows (LUW) or z/OS, so that you access databases that are stored on these platforms with convenience. This chapter shows you how to access the database of DB2 LUW or z/OS.

7.1.1 Preparing the DB2 for LUW or z/OS environment

The first step in accessing other DB2 data is to create a database directory entry on IBM i for the target database on DB2 LUW or z/OS. To add a new relational database directory entry, run the **`ADDRDBDIRE`** CL command. The following list describes the entry attributes:

- Relational database

  Specify the relational database name at the remote location. You can specify a maximum of 18 characters for the name; however, DB2 for z/OS relational database names are limited to 16 characters. The data source that is specified is case-sensitive. Make sure that the name exactly matches what is in WRKRDBDIRE.

- Relational database alias

  Specify the relational database alias. The alias is used for locally identifying the relational database that is specified above. You can specify a maximum of 18 characters for the alias.

- Name or address

  Specifies the remote location name or IP address of the system on which the relational database (RDB) is.

- Type

  Choose *IP.

- Port number or service program.

  For DB2 LUW, specify port 50000 and for z/OS specify port 446, unless you overrode the default configuration to a different port.

The next step is to use a JAVA CL command to create the necessary DB2 CLI packages on the DB2 LUW or z/OS system. From IBM i, run the following command (filling in rdb_alias, user ID, and password):

```
JAVA CLASS(com.ibm.db2.jdbc.app.DB2PackageCreator) PARM('rdb_alias' 'userid'
'password') PROP((jdbc.drivers 'com.ibm.as400.access.AS400JDBCDriver'))
```

> **Note**: Ignore the message "`com.ibm.db2.jdbc.app.DB2DBException: COMMIT(*NONE) valid only if relational database rdb_alias(product identification SQL10056) is DB2 for IBM i`" because neither DB2 LUW or z/OS support the `*NONE` parameter. The **`JAVA`** command handles the error and continues.

After the relational database directory is added successfully, you can verify the data source connection by running the following SQL statements:

1. Run **`STRSQL`** on IBM i.
2. Run **`SQL: CONNECT TO`** ***`databaseName`*** **`USER`** ***`user`*** **`USING '`*****`password`*****`'`**.

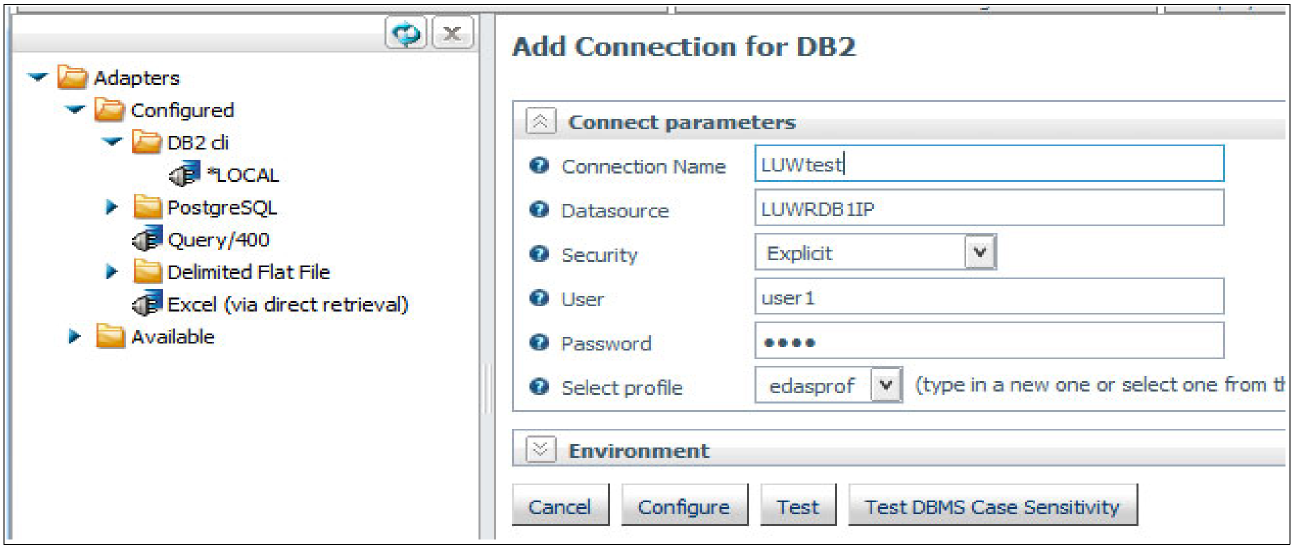
   Specify *databaseName* as **`Relational database alias`**, and the user and password of this database.

The connection is established if it shows **`Current connection is to relational database`** *`datebaseName`*.

7.1.2 Configuring the adapter for DB2 LUW or z/OS

Configuring the DB2 Web Query adapter consists of specifying connection and authentication information for each of the connections you want to establish. To accomplish this task, complete the following steps:

1. Log on to DB2 Web Query as a developer or an administrator.
2. Right-click on a folder in the left panel and select **Metadata,** → **New**. A new window opens.
3. Right-click **DB2 cli** and select **Add connection**. The Add connection for DB2 window opens, as shown in Figure 7-1.



Connection Name: LUWtest
Datasource: LUWRDB1IP
Security: Explicit
User: user1
Password: ••••
Select profile: edasprof (type in a new one or select one from th

Figure 7-1 Add Connection to DB2 for Linux, UNIX, or Windows or z/OS

The following list describes the connection attributes for which you can supply values. You can refer to Figure 7-1.

– Connection name

Logical name that is used to identify this particular set of connection attributes. The default is CON01.

– Datasource

DB2 data source name (DSN). There is no default DSN. You must enter a value. For IBM i, this is the Remote Database Directory entry or *LOCAL (for local host). For DB2 LUW or z/OS, it is the relational database name that is configured in the relational database directory.

- Security

  There are three methods by which a user can be authenticated when connecting to a database server:

  Explicit The user ID and password are explicitly specified for each connection and passed to the database, at connection time, for authentication.

  Password Passthru The user ID and password that is received from the client application are passed to the database, at connection time, for authentication.

  Trusted The adapter connects to the database by using the database rules for an impersonated process that are relevant to the current operating system.

- User

  The primary authorization ID by which you are known to the data source.

- Password

  The password that is associated with the primary authorization ID.

- Select profile

  Select a profile from the drop-down menu to indicate the level of profile in which to store the `CONNECTION_ATTRIBUTES` command. The global profile, `edasprof.prf`, is the default. If you want to create a profile, either a user profile (`user.prf`) or a group profile (if available on your platform and using the appropriate naming convention), select **New Profile** from the drop-down menu and enter a name in the Profile Name field (the extension is added automatically). Store the connection attributes in the server profile (`edasprof`).

4. Click **Configure**.

   If you want to make sure that your configuration is correct, click **Test**. After your configuration is valid, a new web page opens that contains a result set from your new adapter.

After you configure the new adapter, you should restart DB2 Web Query before creating the metadata. Use the `ENDWEBQRY` and `STRWEBQRY` CL commands to restart Web Query.

7.1.3 Creating metadata

After you establish a connection to a data source, you can create various synonyms over the data (tables, views, and stored procedures) at the data source that you want to query by completing the following steps:

1. Right-click on a folder in the left panel and select **Metadata** → **New**.
2. In the Adapter pane, you should now see your DB2 for Linux, UNIX, or Windows or z/OS installation as a data source. Right-click the connection and select **Create or Update Synonym**.
3. Identify the target database on the DB2 LUW or z/OS.
4. Click **Next**.
5. Select the tables that you are interested in querying. Provide a prefix or suffix if desired and click **Create Synonym** after you select the tables of interest. For more information about each parameter, see 7.5.3, “Creating metadata” on page 157.
6. Click **Create Synonym**.

You should see a summary completion window.

You now have a synonym that is specific to tables that are in the database on DB2 LUW or z/OS. You can now use this new synonym to build reports by using InfoAssist or Developer Workbench, the same way that you do with local data synonyms.

7.2 Using the adapter for Microsoft SQL Server

As of this writing, versions of MS SQL Server that are supported are 2005, 2008, and 2012/2014. Due to Microsoft licensing agreements, IBM cannot ship the required code as part of the DB2 Web Query for i package. Therefore, to complete the installation, you must install a version of the Microsoft JDBC driver for SQL Server on your system.

> **Note:** To use the SQL Server adapter, you must be running with DB2 Web Query Standard Edition.

7.2.1 Preparing the Microsoft SQL Server environment

To determine which version to download, Table 7-1 clarifies which driver version and `.jar` file are needed for which DB2 Web Query release. You can find more information about the driver downloads at this Microsoft website.

Table 7-1 Versions that are needed for DB2 Web Query for i

DB2 Web Query for i release	Microsoft JDBC Driver version	Java SE version	OS version	SQL Server version
DB2 Web Query for i 2.1	4.0 - sqljdbc4.jar	6	6.1, 7.1, or 7.2	2008, 2008 R2, 2012, 2014, or 2016
DB2 Web Query for i 2.2	4.1 - sqljdbc41.jar	7	7.1, 7.2, or 7.3	2008, 2008 R2, 2012, 2014, or 2016

At the time of writing, Table 7-1 has the relevant driver information. You can verify the most current information at the DB2 Web Query wiki.

Download the JDBC driver to your desktop and extract the `.jar` file that is required by the adapter. Upload the `.jar` file to the IBM i environment and place it a directory that is accessible to DB2 Web Query. The IBM i Java JVM uses a default extensions directory path for the location of user-defined extensions. The default location can be either of the following:

- `/QIBM/UserData/Java400/ext`
- The Java version-specific JDK location. The Java 1.7 location is shown in this example:

  `/QOpenSys/QIBM/ProdData/JavaVM/jdk70/32bit/jre/lib/ext`

When a JDBC driver file exists in either of these locations, it is loaded by the JVM before any extension is defined in another location. So, you do not need to set **`IBI_CLASSPATH`**. You can also upload the `.jar` file to a directory that you choose, for example:

```
/qibm/UserData/qwebqry/ibi/srv77/wfs/user
```

Then, the **`IBI_CLASSPATH`** variable must be set up as part of the adapter configuration step. The variable is the path of your JDBC .jar file. For example, the `IBI_CLASSPATH` is the following one:

```
/qibm/UserData/qwebqry/ibi/srv77/wfs/user/jdbc.jar
```

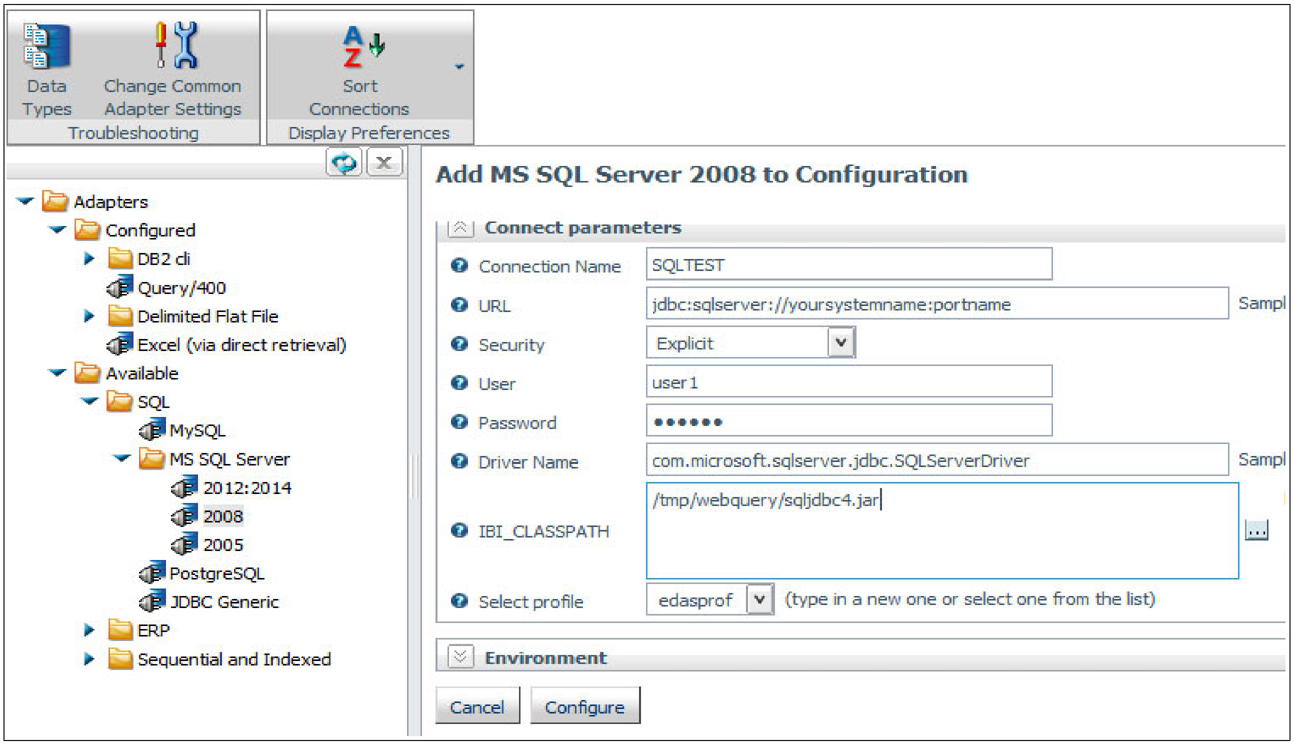
Java version 1.7 or higher is required for DB2 Web Query for i Version 2.2.0.

> **Tips:** To avoid conflicts with other applications or extra work when DB2 Web Query moves to Java 8.0 or later, the preferred practice is to put the drivers into their own extensions directory or specify them through the **`IBI_CLASSPATH`** parameter.

7.2.2 Establishing a connection to your Microsoft SQL Server

To configure the adapter you must provide both connection and authentication information. To do so, complete the following steps:

1. Log on to DB2 Web Query as a developer or an administrator.
2. Right-click on a folder in the left panel and select **Metadata** → **New**. A window opens.
3. Select **Available** → **SQL** → **MS SQL Server**. You see three possible MS SQL Server versions (2012:2014, 2008, and 2005). Choose the version that matches the level of driver that you installed, which might not be the same as the level of SQL Server to which you are connecting. For this example, right-click **2008 (Unicode Optional)** and select **Configure**. The Add MS SQL Server 2008 to Configuration window opens, as shown in Figure 7-2.



Add MS SQL Server 2008 to Configuration

Connect parameters

Connection Name: SQLTEST
URL: jdbc:sqlserver://yoursystemname:portname
Security: Explicit
User: user1
Password: ••••••
Driver Name: com.microsoft.sqlserver.jdbc.SQLServerDriver
IBI_CLASSPATH: /tmp/webquery/sqljdbc4.jar
Select profile: edasprof (type in a new one or select one from the list)

Figure 7-2 Add MS SQL Server 2008 to the configuration

4. In the configuration window, name your adapter. In this example, the connection name is called SQLTEST, but it can be any name that you want.

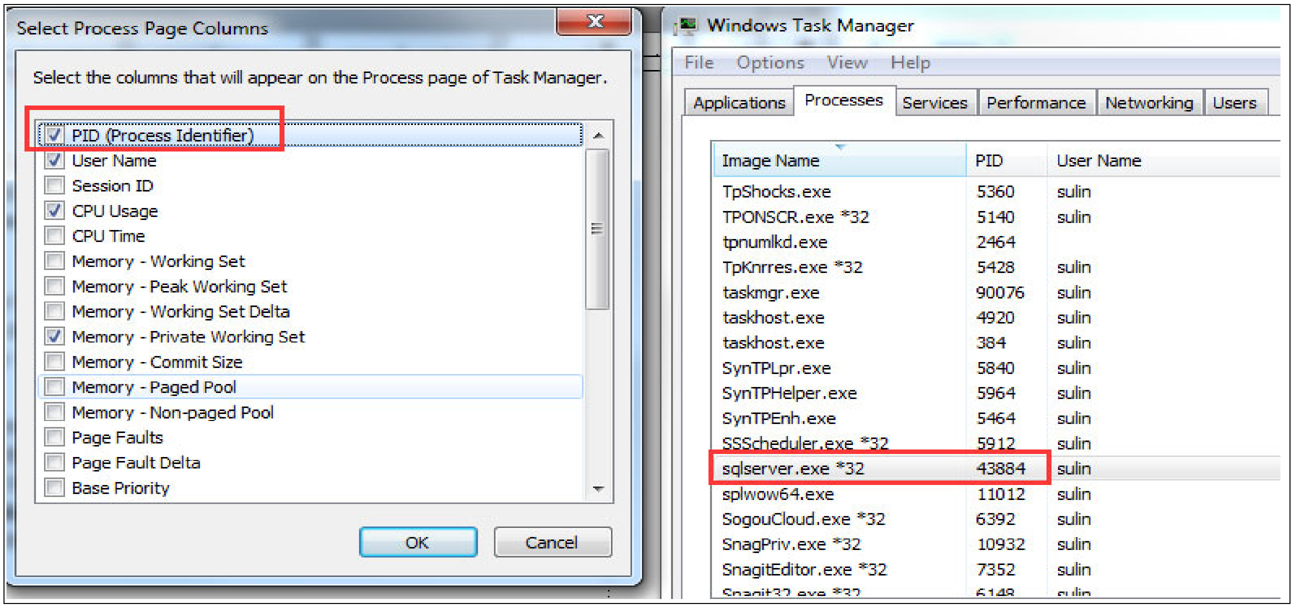
5. Enter the URL location for your SQL Server data source, which is the driver prefix, the separator (://), followed by the host name (location) of the SQL Server, the separator (:), and the port, in the following format:

   ```
   prefix://hostname:port
   ```

   In this example, the prefix is `jdbc:sqlserver`. The host name for your system can be either the DNS name (`sqlserver.mycompany.com`) or an IP address (`1.2.3.4`). If you use a DNS name, make sure that IBM i can resolve to it by doing a simple **`PING`** from the CL command line to the SQL Server DNS name.

   The final piece of this URL is the port number. The default port for the SQL Server is 1433. However, not all SQL Servers run on that port. To find your SQL Server's port number, complete the following steps on the SQL Server machine:

   a. Open the Windows Task Manager and add the Process ID (PID) column by clicking **View** → **Select Columns** → **Check the PID option** → **OK**. Find the image name `sqlservr.exe` and record its PID, as shown in Figure 7-3.



Select the columns that will appear on the Process page of Task Manager.

- ☑ PID (Process Identifier)
- ☑ User Name
- □ Session ID
- ☑ CPU Usage
- □ CPU Time
- □ Memory - Working Set
- □ Memory - Peak Working Set
- □ Memory - Working Set Delta
- ☑ Memory - Private Working Set
- □ Memory - Commit Size
- □ Memory - Paged Pool
- □ Memory - Non-paged Pool
- □ Page Faults
- □ Page Fault Delta
- □ Base Priority

Image Name	PID	User Name
TpShocks.exe	5360	sulin
TPONSCR.exe *32	5140	sulin
tpnumlkd.exe	2464	
TpKnrres.exe *32	5428	sulin
taskmgr.exe	90076	sulin
taskhost.exe	4920	sulin
taskhost.exe	384	sulin
SynTPLpr.exe	5840	sulin
SynTPHelper.exe	5964	sulin
SynTPEnh.exe	5464	sulin
SSScheduler.exe *32	5912	sulin
sqlserver.exe *32	43884	sulin
splwow64.exe	11012	sulin
SogouCloud.exe *32	6392	sulin
SnagPriv.exe *32	10932	sulin
SnagitEditor.exe *32	7352	sulin
Snagit32.exe *32	6148	sulin

Figure 7-3 Get the PID number for SQL Server

   b. Open a command prompt, run **`netstat -a -n -o`**, and look for the matching PID number. Under the Local Address column is the entry `0.0.0.0:xyz`. The value `xyz` after the colon is the port number.

   Assume that the DNS name of the SQL Server is `sqltest.rchland.ibm.com`, the port is 1433, and the version of the driver is 2.0. The URL would then be the following one:

   ```
   jdbc:sqlserver://sqltest.rchland.ibm.com:1433
   ```

6. There are three methods by which a user can be authenticated when connecting to a Microsoft SQL Server:

   Explicit The user ID and password are explicitly specified for each connection and passed to Microsoft SQL Server, at connection time, for authentication as a standard login. This option requires that SQL Server security is set to SQL Server and Windows (for Windows), or else to SQL Server and UNIX.

   Password Passthru (Windows only) The user ID and password that is received from the client application are passed to Microsoft SQL Server, at connection time, for authentication as a standard login. This option requires that SQL Server security is set to SQL Server and Windows.

   Trusted The adapter connects to Microsoft SQL Server as an operating system login by using the credentials of the operating system user that is impersonated by the server data access agent. This option works with either of the SQL Server security settings.

7. The final step is to enter the driver name. Enter `com.microsoft.sqlserver.jdbc.SQLServerDriver` for the driver name.
8. Do not be concerned that the CLASSPATH is NOT SET, provided that you posted the JDBC driver `.jar` file in the above directory location. DB2 Web Query will find the necessary driver.
9. If you want to make sure that your configuration is correct, click **Test**. If your configuration is valid, a new web page opens containing a result set from your SQL Server showing `Microsoft SQL Server ADAPTER TEST COMPLETED SUCCESSFULLY!`
10. Click **Configure** after the appropriate information is provided. You should receive a message stating that you about to change a server’s configuration.

Now that you have configured the new adapter, you should restart DB2 Web Query before moving on to create metadata. Use the **`ENDWEBQRY`** and **`STRWEBQRY`** commands to restart.

7.2.3 Creating metadata

After you establish a connection to a data source, you can create various synonyms over the data (tables, views, and stored procedures) at the data source that you want to query. Complete the following steps:

1. Right-click on a folder in the left panel and select **Metadata** → **New**.
2. In the Adapter pane, you should now see your MS SQL Server as a data source. Right-click the newly created connection and select **Create Synonym or Update synonym**.
3. Identify the target database on the SQL Server. Remember, SQL Server is different from IBM i because it has the concept of multiple databases. Within each of those databases are the schemas that contain the tables.
4. Click **Next**.
5. Select the tables that you are interested in querying. Provide a prefix or suffix if you want and click **Create Synonym** after you select the tables of interest. For more information about each parameter, see 4.2.1, “DB2 Web Query for i Metadata interfaces” on page 60.
6. Click **Create Synonym**. You should see a summary window.

You now have a synonym that is specific to tables that are in a database on an MS SQL Server. You can now use this new synonym to build reports by using InfoAssist, or Developer Workbench, the same way that you do with local data synonyms.

7.3 Using the adapter for MySQL

The adapter for MySQL allows applications to access MySQL data sources. The adapter converts application requests into native MySQL statements and returns optimized answer sets to the requesting application.

> **Note:** To use the MySQL adapter, you must be running with DB2 Web Query Standard Edition.

7.3.1 Preparing the MySQL environment

Download the MySQL or MariaDB JDBC driver to your desktop and extract the necessary `.jar` file that is required by the adapter. Upload the `.jar` file to the IBM i environment and place it in a directory that is accessible to DB2 Web Query. The IBM i Java JVM uses a default extension directory path for the user-defined extensions. The default location can be either of the following options:

- `/QIBM/UserData/Java400/ext`
- The Java version-specific JDK location. The Java 1.7 location is shown in this example:

  `/QOpenSys/QIBM/ProdData/JavaVM/jdk70/32bit/jre/lib/ext`

When a JDBC driver file exists in either of these locations, it is loaded by the JVM before any extension that is defined in another location. So, you do not need to set **`IBI_CLASSPATH`**.

You can also upload the `.jar` file to a directory, for example:

```
/qibm/UserData/qwebqry/ibi/srv77/wfs/user
```

In this case, the **`IBI_CLASSPATH`** variable must be set up as part of the adapter configuration step. The variable is the path of your JDBC `.jar` file. For example, IBI_CLASSPATH is `/qibm/UserData/qwebqry/ibi/srv77/wfs/user/jdbc.jar`.

Java version 1.7 or higher is required for DB2 Web Query for i Version 2.2.0.

> **Tips:** To avoid conflicts with other applications or extra work when DB2 Web Query moves to Java 8.0 or later, the preferred practice is to put the drivers into their own extensions directory or specify them through the **`IBI_CLASSPATH`** parameter.

7.3.2 Configuring the adapter for MySQL

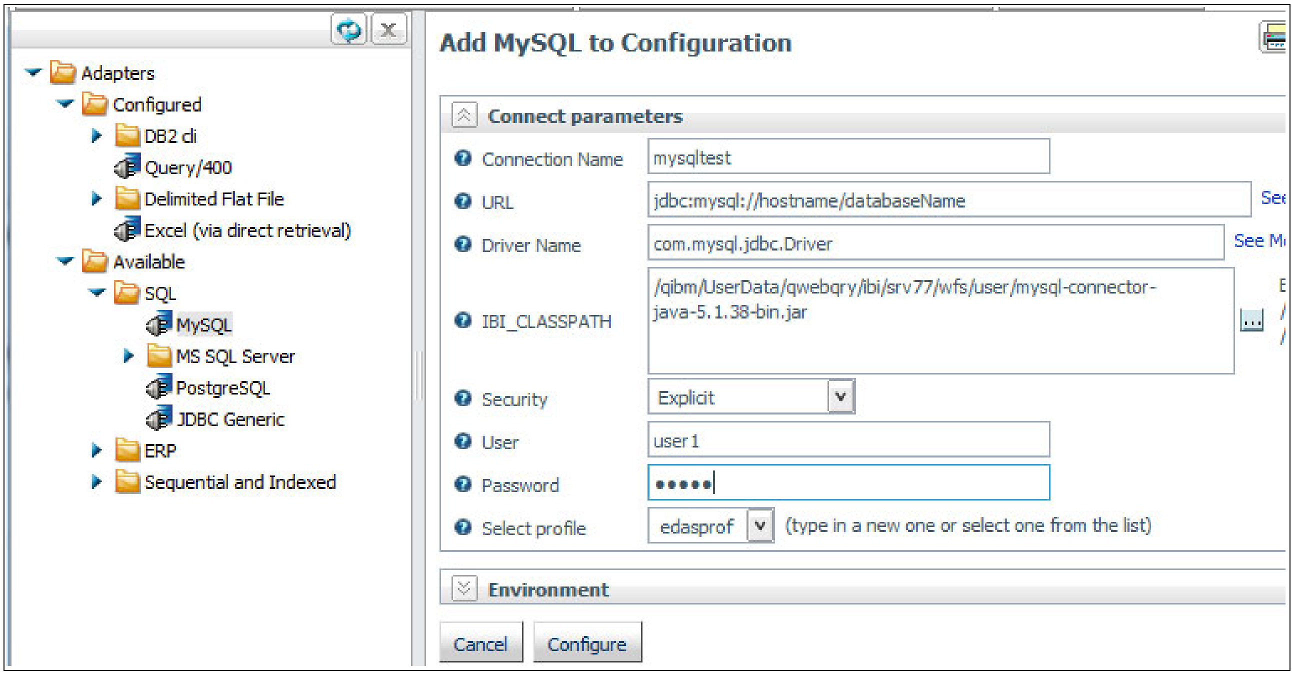
Configuring the adapter consists of specifying connection and authentication information for each of the connections that you want to establish. The MySQL adapter is under the `SQL` group folder. Complete the following steps:

1. Log in as a developer or an administrator.
2. Right-click on a folder in the left panel and select **Metadata** → **New**. A window opens. From this window, expand **Available** → **SQL**.

3. Right-click **MySQL** and select **Configure**. You should see the Add MySQL to Configuration window.
4. The following list describes the connection attributes for which you can supply values, as shown in Figure 7-4:

   – Connection name

   The logical name that is used to identify this particular set of connection attributes. The default is CON01. Here, we rename it to mysqltest.



Add MySQL to Configuration

Connect parameters

Connection Name: mysqltest

URL: jdbc:mysql://hostname/databaseName

Driver Name: com.mysql.jdbc.Driver

IBI_CLASSPATH: /qibm/UserData/qwebqry/ibi/srv77/wfs/user/mysql-connector-java-5.1.38-bin.jar

Security: Explicit

User: user1

Password: •••••

Select profile: edasprof (type in a new one or select one from the list)

Figure 7-4 Add MySQL to configuration

   – URL

   Enter the location URL for the MySQL data source. The basic syntax is as follows:

   ```
   jdbc:mysql://host/database
   ```

   Where:

   - `host` is the computer name or IP address where the MySQL database is.
   - `database` is the name of the database.

   Here are two examples:

   - `jdbc:mysql://localhost/qatst`
   - `jdbc:mysql://edaaix52/qatst`

   Beginning with MySQL Release 4.1, you can reference additional MySQL connection properties in the URL. The first property must be preceded by the ? character, and the second and subsequent properties that are referenced in the URL must be preceded by the & character followed immediately by an | (pipe) character, as illustrated in the following example.

   Suppose that you want to add the following connection properties:

   - `sessionVariables=sql_mode=PIPES_AS_CONCAT`
   - `zeroDateTimeBehavior=convertToNull`

Enter the following URL and properties:

```
jdbc:mysql://host/database?sessionVariables=sql_mode=P
IPES_AS_CONCAT&|
zeroDateTimeBehavior=convertToNull
```

Note: The URL must be entered as a single line, without a space after the | character.

- Driver name

  The name of the JDBC driver:

  `com.mysql.jdbc.Driver`

- IBI_CLASSPATH

  Defines the additional Java Class directories or full-path `.jar` file names that are available for Java Services. You can enter one reference per line.

- Security

  There are two methods by which a user can be authenticated when connecting to a database server:

  Explicit The user ID and password are explicitly specified for each connection and passed to the database, at connection time, for authentication.

  Password Passthru The user ID and password that are received from the client application are passed to the database, at connection time, for authentication.

- User

  The primary authorization ID by which you are known to the data source.

- Password

  The password that is associated with the primary authorization ID.

- Select profile

  Select a profile from the drop-down menu to indicate the level of profile in which to store the **`CONNECTION_ATTRIBUTES`** command. The global profile, `edasprof.prf`, is the default. If you want to create a profile, either a user profile (*`user`*`.prf`) or a group profile (if available on your platform by using the appropriate naming convention), select **New Profile** from the drop-down menu and enter a name in the Profile Name field (the extension is added automatically). Store the connection attributes in the server profile (`edasprof`).

5. Click **Configure**.

   If you want to make sure that your configuration is correct, click **Test**. After your configuration is valid, a new web page opens and contains a result set from your new adapter.

After you configure the new adapter, you should restart Web Query before you create metadata. Use the **`ENDWEBQRY`** and **`STRWEBQRY`** commands to restart.

7.3.3 Creating metadata

After you establish a connection to a data source, you can create various synonyms over the data (tables, views, and stored procedures) at the data source that you want to query. Complete the following steps:

1. Right-click on a folder in the left panel and select **Metadata** → **New**.
2. In the Adapter pane, you should now see your MySQL as a data source. Right-click the newly created connection and select **Create or Update Synonym**.
3. Identify the target database on the MySQL. Remember, MySQL is different from IBM i in that it uses the concept of multiple databases. Within each of those databases are the schemas that contain the tables.
4. Click **Next**.
5. Select the tables that you are interested in querying. Provide a prefix or suffix if you want and click **Create Synonym** after you select the tables of interest. For more information about each parameter, see 7.5.3, “Creating metadata” on page 157.
6. Click **Create Synonym**. You see a summary completion window.

You now have a synonym that is specific to tables in a database on MySQL. You can use this new synonym to build reports by using InfoAssist, or Developer Workbench, the same way that you do with local data synonyms.

7.4 Using the adapter for PostgreSQL

The adapter for PostgreSQL allows applications to access PostgreSQL data sources. The adapter converts application requests into PostgreSQL calls and returns optimized answer sets to the requesting application.

Note: To use the PostgreSQL adapter, you must be running with DB2 Web Query Standard Edition.

7.4.1 Preparing the PostgreSQL environment

To use the adapter for PostgreSQL, you must install the PostgreSQL driver, set its **IBI_CLASSPATH** value before server startup, and ensure that the JSCOM3 service is running. Download the Postgresql JDBC driver to your desktop and extract the necessary `.jar` file that is required by the adapter. Upload the `.jar` file to the IBM i environment and place it a directory that is accessible to DB2 Web Query. The IBM i Java JVM uses a default extensions directory path for the location of use-defined extensions. The default location can be either of the following options:

- `/QIBM/UserData/Java400/ext`
- The Java version-specific JDK location. The Java 1.7 location is shown in this example:

  `/QOpenSys/QIBM/ProdData/JavaVM/jdk70/32bit/jre/lib/ext`

When a JDBC driver file exists in either of these locations, it is loaded by the JVM before any extension is defined in another location. So, you do not need to set **IBI_CLASSPATH**. You can also upload the `.jar` file to a directory of your choosing, for example:

```
/qibm/UserData/qwebqry/ibi/srv77/wfs/user
```

Then, the **`IBI_CLASSPATH`** variable must be set up as part of the adapter configuration step. The variable is the path of your JDBC `.jar` file. For example, the IBI_CLASSPATH is `/qibm/UserData/qwebqry/ibi/srv77/wfs/user/jdbc.jar`.

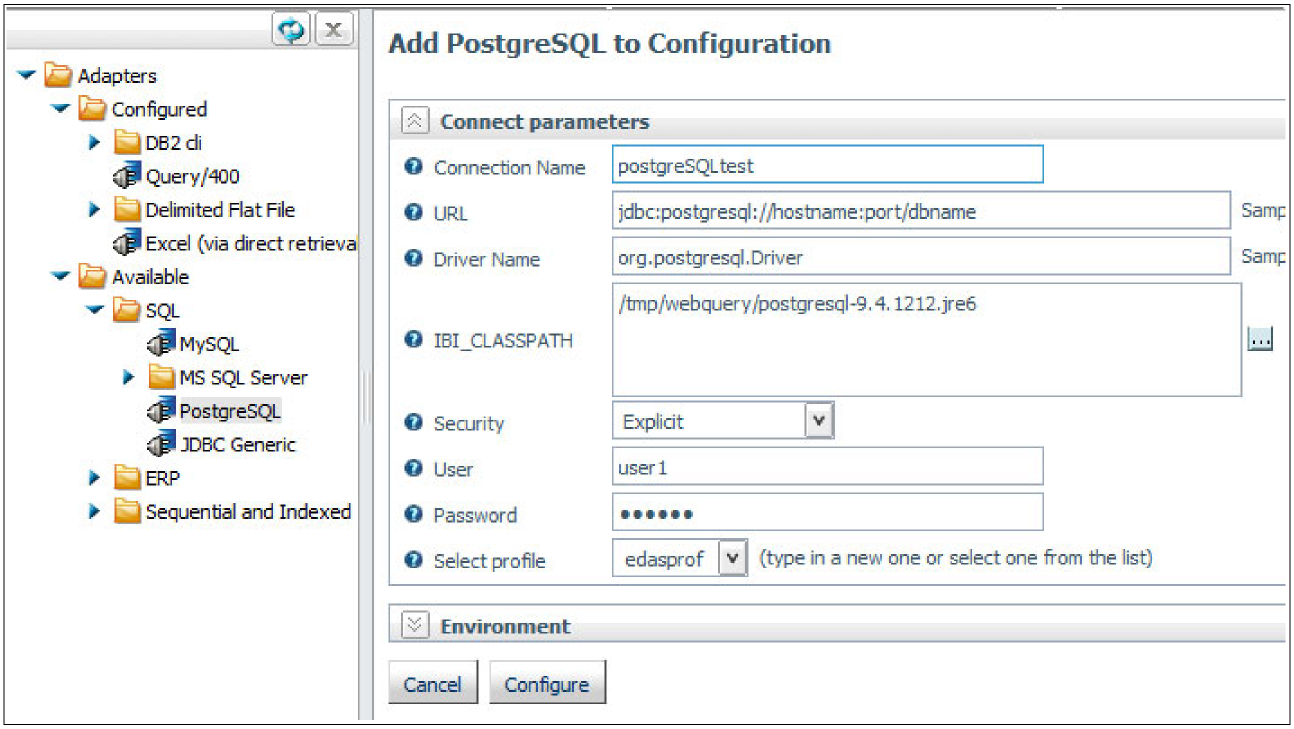
Java version 1.7 or higher is required for DB2 Web Query Version 2.2.0.

> **Tips:** To avoid conflicts with other applications or extra work when DB2 Web Query moves to Java 8.0 or later, the preferred practice is to put the drivers into their own extensions directory or specify them through the **`IBI_CLASSPATH`** parameter.

7.4.2 Configuring the adapter for PostgreSQL

Configuring the adapter consists of specifying connection and authentication information for each of the connections that you want to establish. The PostgreSQL adapter is under the `SQL` group folder. To complete the attribute declaration, click **Configure**. Complete the following steps:

1. Log in as a developer or an administrator.
2. Right-click on a folder in the left panel and select **Metadata,** → **New**. A window opens. From this window, expand **Available** → **SQL**.
3. Right-click **PostgreSQL** and select **Configure**. You see the Add PostgreSQL to configuration window opens. as shown in Figure 7-5.



Add PostgreSQL to Configuration

Connect parameters

Connection Name: postgreSQLtest

URL: jdbc:postgresql://hostname:port/dbname Samp

Driver Name: org.postgresql.Driver Samp

IBI_CLASSPATH: /tmp/webquery/postgresql-9.4.1212.jre6

Security: Explicit

User: user1

Password: ••••••

Select profile: edasprof (type in a new one or select one from the list)

Figure 7-5 Add PostgreSQL to Configuration

The following list describes the connection attributes for which you can supply values:

– Connection name

The logical name that is used to identify this particular set of connection attributes. The default is CON01.

- URL

  The Location URL for the PostgreSQL data source.

- Driver name

  The name of the PostgreSQL JDBC driver, for example, `org.postgresql.Driver`. See the PostgreSQL documentation for the specific driver release that you are using.

- IBI_CLASSPATH

  Defines the additional Java Class directories or full-path `.jar` file names that are available for Java Services. You can enter one reference per line.

- Security

  There are two methods by which a user can be authenticated when connecting to a database server:

  Explicit The user ID and password are explicitly specified for each connection and passed to the database, at connection time, for authentication.

  Password Passthru The user ID and password that are received from the client application are passed to the database, at connection time, for authentication.

- User

  The primary authorization ID by which you are known to the data source.

- Password

  The password that is associated with the primary authorization ID.

- Select profile

  Select a profile from the drop-down menu to indicate the level of profile in which to store the **`CONNECTION_ATTRIBUTES`** command. The global profile, `edasprof.prf`, is the default. If you want to create a profile, either a user profile (*`user`*`.prf`) or a group profile (if available on your platform by using the appropriate naming convention), choose **New Profile** from the drop-down menu and enter a name in the Profile Name field (the extension is added automatically). Store the connection attributes in the server profile (`edasprof`).

4. Click **Configure**.

   If you want to make sure that your configuration is correct, click **Test**. After your configuration is valid, a new web page opens that contains a result set from your new adapter.

After you configure the new adapter, you should restart the DB2 Web Query before moving on to create metadata. Use the **`ENDWEBQRY`** and **`STRWEBQRY`** commands.

7.4.3 Creating metadata

After you establish a connection to a data source, you can create various synonyms over the data (tables, views, and stored procedures) at the data source that you want to query. Complete the following steps:

1. Right-click on a folder in the left panel and select **Metadata** → **New**.
2. In the Adapter pane, you should now see PostgreSQL as a data source. Right-click the newly created connection and select **Create or Update Synonym**.
3. Identify the target database on the PostgreSQL.

4. Click **Next**.
5. Select the tables that you are interested in querying. Provide the prefix or suffix if you want one and click **Create Synonym** after you have selected the tables of interest. For more information about each parameter, see 7.5.3, “Creating metadata” on page 157.
6. Click **Create Synonym**. You see a summary completion window.

You now have a synonym that is specific to tables located in a database on an PostgreSQL. You can now use this new synonym to build reports by using InfoAssist, or Developer Workbench, the same way that you do with local data synonyms.

7.5 Using the generic adapter for JDBC

This section describes how to configure the *generic* adapter for JDBC. This adapter for JDBC allows applications to access specific JDBC data sources beyond the specific adapters that are described earlier in this chapter. For example, if you want to connect to Oracle, you can acquire an Oracle JDBC driver and leverage this generic adapter to access data in Oracle. The adapter converts application requests into JDBC calls and retunes optimized answer sets to the requesting application..

> **Note:** To use the generic JDBC adapter, you must be running with DB2 Web Query Standard Edition.

7.5.1 Setting up JDBC environment

Download the JDBC driver for your database (not included with DB2 Web Query) to your desktop and extract the necessary `.jar` file that is required by the adapter. Upload the `.jar` file to the IBM i environment and place it a directory that is accessible to DB2 Web Query. The IBM i Java JVM uses a default extensions directory path for the location of user-defined extensions. The default location can be either of the following options:

- `/QIBM/UserData/Java400/ext`
- The Java version-specific JDK location. The Java 1.7 location is shown in this example:

  `/QOpenSys/QIBM/ProdData/JavaVM/jdk70/32bit/jre/lib/ext`

When a JDBC driver file exists in either of these locations, it is loaded by the JVM before any extension is defined in another location.

You can also download the `.jar` file to a directory of your preference, for example:

```
/qibm/UserData/qwebqry/ibi/srv77/wfs/user
```

In this case, the **`IBI_CLASSPATH`** variable must be set up as part of the adapter configuration step. The variable is the path of your JDBC jar file. In this example, IBI_CLASSPATH is as follows:

```
/qibm/UserData/qwebqry/ibi/srv77/wfs/user/jdbc.jar
```

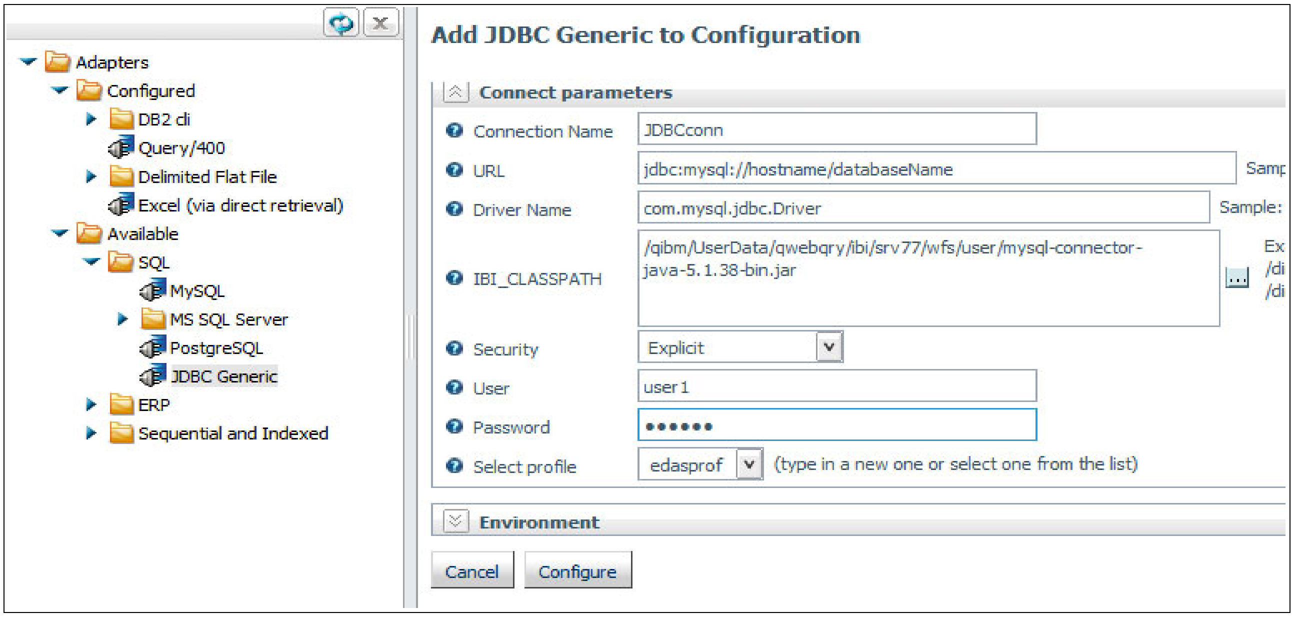
Java version 1.7 or higher is required for DB2 Web Query Version 2.2.0.

> **Tips:** To avoid conflicts with other applications or extra work when DB2 Web Query moves to Java 8.0 or later, the preferred practice is to put the drivers into their own extensions directory or specify them through the **`IBI_CLASSPATH`** parameter.

7.5.2 Configuring the adapter for JDBC

Configuring the adapter consists of specifying connection and authentication information for each of the connections you want to establish. Complete the following steps:

1. Log in using a user ID that is assigned as an Administrator (but not the QWQADMIN user ID).
2. Right-click on a folder in the left panel and select **Metadata** → **New**.
3. Click **Adapters**, and then in the left pane click **Available** → **SQL**.
4. Right-click **JDBC Generic** and select **Configure**. You see the Add JDBC Generic to Configuration window. The following list describes the connection attributes that you need to fill out, as shown in Figure 7-6. Right-click **JDBC Generic** and select **Configure**. You see the Add JDBC Generic to Configuration window, as shown in Figure 7-6.



Add JDBC Generic to Configuration

Connect parameters

Connection Name: JDBCconn

URL: jdbc:mysql://hostname/databaseName Samp

Driver Name: com.mysql.jdbc.Driver Sample:

IBI_CLASSPATH: /qibm/UserData/qwebqry/ibi/srv77/wfs/user/mysql-connector-java-5.1.38-bin.jar Ex /di /di

Security: Explicit

User: user1

Password: ••••••

Select profile: edasprof (type in a new one or select one from the list)

Environment

Figure 7-6 Add a JDBC connection

The following list describes the connection attributes. The default is CON01, but you can change it to whatever you want.

- Connection name:

  The logical name that is used to identify this particular set of connection attributes. The default is CON01, but you can change it to whatever you want.

- URL

  The location URL for the JDBC data source.

- Driver name

  The name for the JDBC driver.

- IBI_CLASSPATH

  Defines the additional Java Class directories or full-path `.jar` file names that are available for Java Services. You can enter one reference per line.

– Security

There are two methods by which a user can be authenticated when connecting to a database server:

Explicit The user ID and password are explicitly specified for each connection and passed to the database, at connection time, for authentication.

Password Passthru The user ID and password that are received from the client application are passed to the database, at connection time, for authentication.

– User

The primary authorization ID by which you are known to the data source.

– Password

The password that is associated with the primary authorization ID.

When you are, click **Configure**. You should be able to see whether the connection is created successfully. If you want to make sure that your configuration is correct, click **Test**. Once your configuration is valid, a new web page opens that contains a result set from your new adapter.

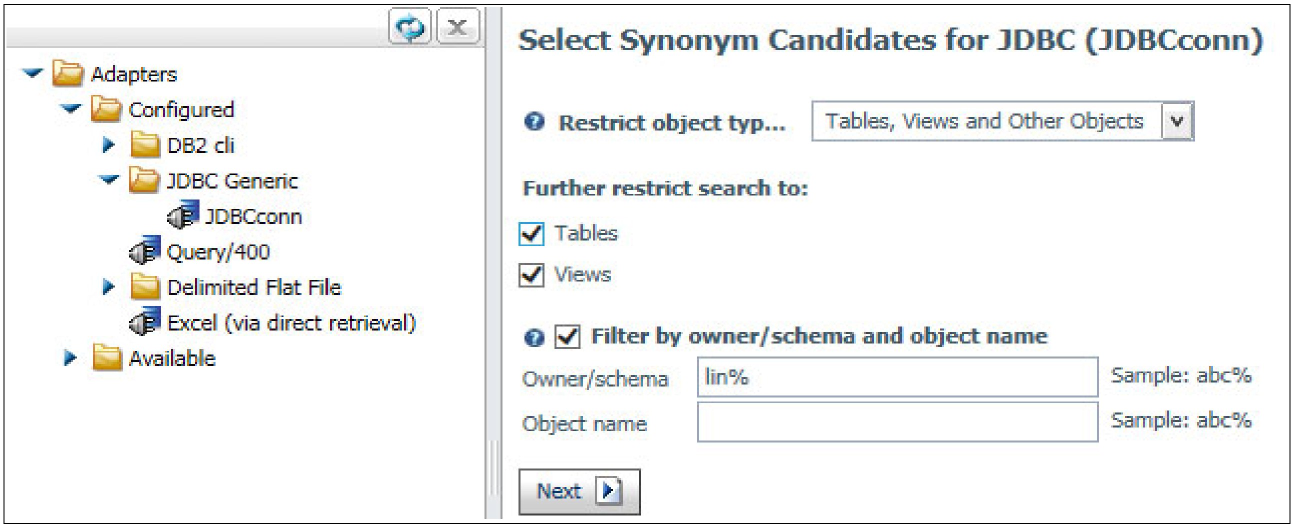
5. Click **Configure** when the appropriate information is provided.

After you configure the new adapter, you should restart DB2 Web Query before creating the metadata. Use the `ENDWEBQRY` and `STRWEBQRY` commands to restart.

7.5.3 Creating metadata

To create a synonym (which is the DB2 Web Query for i term for a metadata object), you must have previously configured the adapter. You can create a synonym from the Applications or Adapters pages of the Web Console. Complete the following steps:

1. Log on to the DB2 Web Query BI portal. Right-click on a folder in the left panel and select **Metadata** → **New**.
2. Expand the **JDBC Generic** folder, right-click the newly created connection, and select **Create or Update Synonym**. Figure 7-7 shows that the filter that is being used is schemas that start with LIN.



Select Synonym Candidates for JDBC (JDBCconn)

Restrict object typ... Tables, Views and Other Objects

Further restrict search to:

☑ Tables

☑ Views

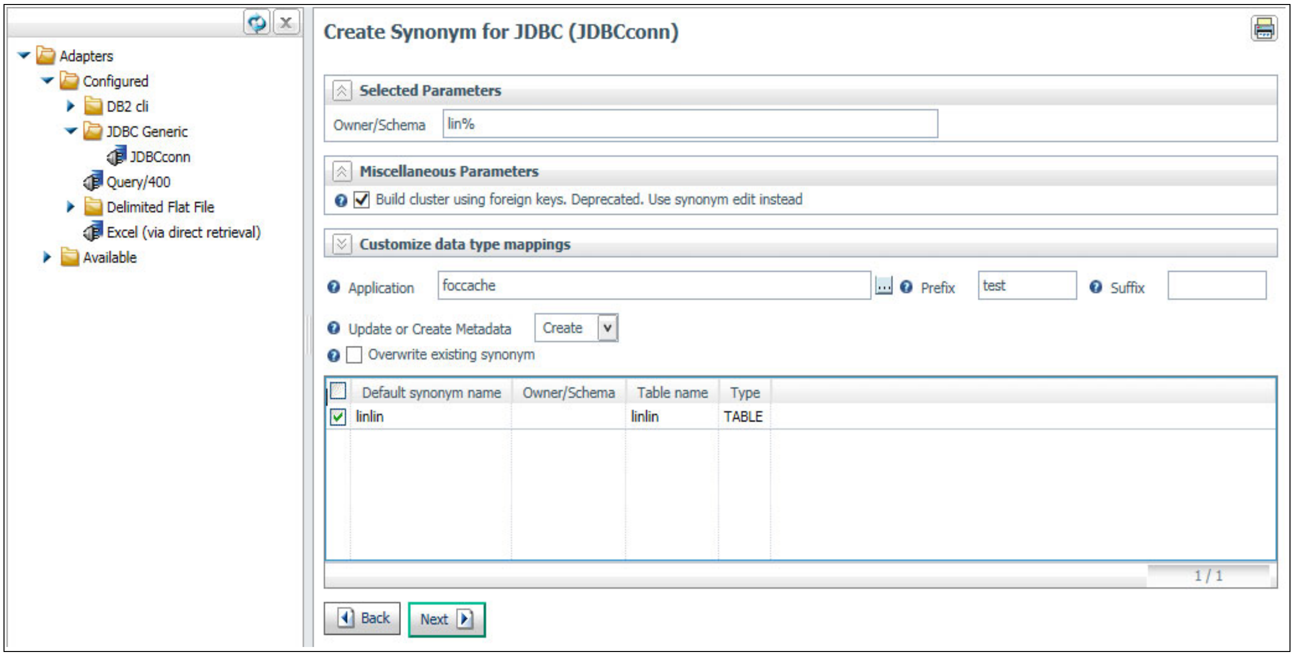
☑ Filter by owner/schema and object name

Owner/schema: lin% Sample: abc%

Object name: Sample: abc%

Figure 7-7 Select synonym for JDCBC

3. The following list describes the parameters for which you must supply values, and the related tasks that you must complete to create a synonym for the adapter. These options might appear on multiple panes. To advance from pane to pane, click the buttons, ending with the **Create Synonym** button, which generates the synonym based on your entries, as shown in Figure 7-8.



Create Synonym for JDBC (JDBCconn)

Selected Parameters

Owner/Schema: lin%

Miscellaneous Parameters

☑ Build cluster using foreign keys. Deprecated. Use synonym edit instead

Customize data type mappings

Application: foccache Prefix: test Suffix:

Update or Create Metadata: Create

☐ Overwrite existing synonym

☐	Default synonym name	Owner/Schema	Table name	Type
☑	linlin		linlin	TABLE

1 / 1

Figure 7-8 Select synonym candidates for JDBC

- Restrict Object Type to

  Restrict candidates for synonym creation based on the selected object types: Tables, Views, Aliases, and MQTs, and any other supported objects. Choosing **External SQL Scripts** from the drop-down list enables you to represent an SQL Query as a synonym for read-only reporting. A synonym candidate can be any file that contains one (and only one) valid SQL Query and does not contain end-of-statement delimiters and comments. Depending on the adapter, you can further restrict your search by selecting check boxes for the listed objects.

- Library

  Type a string for filtering the Library (or DB2 Collection), inserting the wildcard character (%) as needed at the beginning or end of the string. For example, enter ABC% to select tables or views whose owner IDs begin with the letters ABC, %ABC to select tables or views whose owner IDs end with the letters ABC, or %ABC% to select tables or views whose owner IDs contain the letters ABC at the beginning, middle, or end.

- Object Name.

  Type a string for filtering the table, view, or object names, inserting the wildcard character (%) as needed at the beginning or end of the string.

> **Note:** When you create a synonym for DB2 for i, standard IBM i naming conventions apply to the target data source. Therefore, the Adapter for DB2 supports the use of double-quotation marks around any library name or file name that contains lowercase or NLS characters.

4. Click **Next**. Here are the listed objects:

   – Build a cluster by using foreign keys (deprecated).

     You can select the build cluster by selecting the Foreign keys check box to include within this synonym every table that is related to the current table by a foreign key. However, this option has been deprecated because the preferred way to create a cluster is by using the Synonym Editor. The resulting multi-table synonym describes all of the foreign key relationships of this table.

   – One-part name.

     The One-part name check box is a way to control whether a fully qualified library name is used by the synonym, or if a library list should always be used. The unchecked behavior generates a table name that includes the explicit name of the library containing the table. For example, if you specified a library on the first Create Synonym pane, a qualified name like the following one is automatically created in the Access File:

     ```
     TABLENAME=MYLIB/MYTABLE
     ```

     With this explicit type of entry the library is directly located and searched for the table name. If you select the check box, the library name is ignored and a user's library list controls the search for the table name.

     It is a preferred practice to select this check box to allow for more flexible library list control when reports are run. Also, learn about runtime environments as a way to control library lists at run time as a user option.

   – For Subquery.

     Only available when External SQL Scripts is selected from the Restrict objects type to drop-down menu. When selected, a **`SUBQUERY`** keyword is added to the Access File of the generated synonym. If the corresponding SQL string has valid syntax that can be used in the **`FROM`** statement of the generated SQL (what is known as a Derived Table), then the SQL SCRIPT is processed as a subquery that is embedded into a **`FROM`** clause. This usage allows for more flexibility. For example, the synonym can be used as a target for a JOIN. If the SQL **`SCRIPT`** has parameter markers such as ? or :, or the syntax contains constructs that are invalid for a derived table, for example, ORDER BY, then this keyword should not be selected. At run time, if `SUBQUERY=Y` is present and it is determined that the SQL **`SCRIPT`** cannot be used in the **`FROM`** statement, the setting is ignored, and an FOC1782 warning message is issued. The default is selected (`SUBQUERY=Y`).

   – Application.

     Select an application directory. The default value is baseapp.

   – Prefix/Suffix.

     Optional, but good for identifying synonyms that you create by the type of synonym, or if you have tables with identical table names, you can assign a prefix or a suffix to distinguish them. For example, if you have identically named human resources and payroll tables, assign the prefix HR to distinguish the synonyms for the human resources tables. The resulting synonym name cannot exceed 64 characters. If all tables and views have unique names, leave the prefix and suffix fields blank.

– Customize data type mappings.

To change the data type mappings from their default settings, select this check box. The customizable mappings are displayed.

– Overwrite existing synonyms.

To specify that this synonym should overwrite an existing synonym with the same fully qualified name, select the **Overwrite existing synonyms** check box.

Note: The connected user must have operating system write privileges to re-create a synonym.

Select the check box next to the tables for which you want to create synonyms.

5. Click **Create Synonym**. You see a summary completion window.

You now have a synonym that is specific to tables in a database. You can now use this new synonym to build reports by using InfoAssist, or Developer Workbench, the same way that you do with local data synonyms.

7.6 Using application adapters for JD Edwards

One of the biggest challenges with querying ERP databases is digging out the hidden meaning that is buried in the ERP database model and data dictionary. With the DB2 WebQuery integration with JD Edwards EnterpriseOne andWorld databases you can be shielded from those complexities, standardize data attributes such as default formatting of date or monetary fields, and preserve who is authorized to view the data through the ERP system security model.

DB2 Web Query automatically handles decimal and date conversions, provides meaningful column names, and simplifies the data to report authors and end users. Enhance the metadata to define multi-dimensional relationships for OLAP processing or build derived fields that are standardized for every report.

This section highlights the adapters for Oracle JD Edwards World and Oracle JD Edwards EnterpriseOne (E1) when connecting to a local DB2 for i based JD Edwards (JDE) database. Later in this chapter is information on setup required to connect to a JDE instance on a different LPAR or Server than where DB2 Web Query is installed, for example, if you were wanting to query the JDE data that resides on a DR or HA LPAR or Server.

Note: To use the JDE adapter, both DB2 Web Query Standard Edition AND the JDE adapter must be installed.

7.6.1 Using the adapter for JD Edwards World

The application adapter for JDE World allows DB2 Web Query to access JDE World data sources. With this adapter, data in the JD Edwards World DBMS is displayed by using rules that are contained in dictionary files, ensuring that valid information is returned to the requesting program.

Preparing the JD Edwards World environment

Although no environment preparation steps are required, ensure that your system complies with all software specifications.

Do not add the JDE World adapter to DB2 Web Query if it already contains a configured adapter for JDE EnterpriseOne. If you want to add this adapter to that server configuration, you must remove the existing adapter for JDE EnterpriseOne first.

Overview of the setup process

The setup process is composed of the following basic steps:

1. Configure the adapter for JDE World. Define what type of JDE security to implement.
2. Refresh the metadata repository. You must perform this step initially, and repeat it only if there are changes in the JDE metadata tables. These changes occur infrequently at most sites.
3. Create the JDE World synonyms. Synonyms are required for reporting against a data source.

Configuring the adapter for JD Edwards World

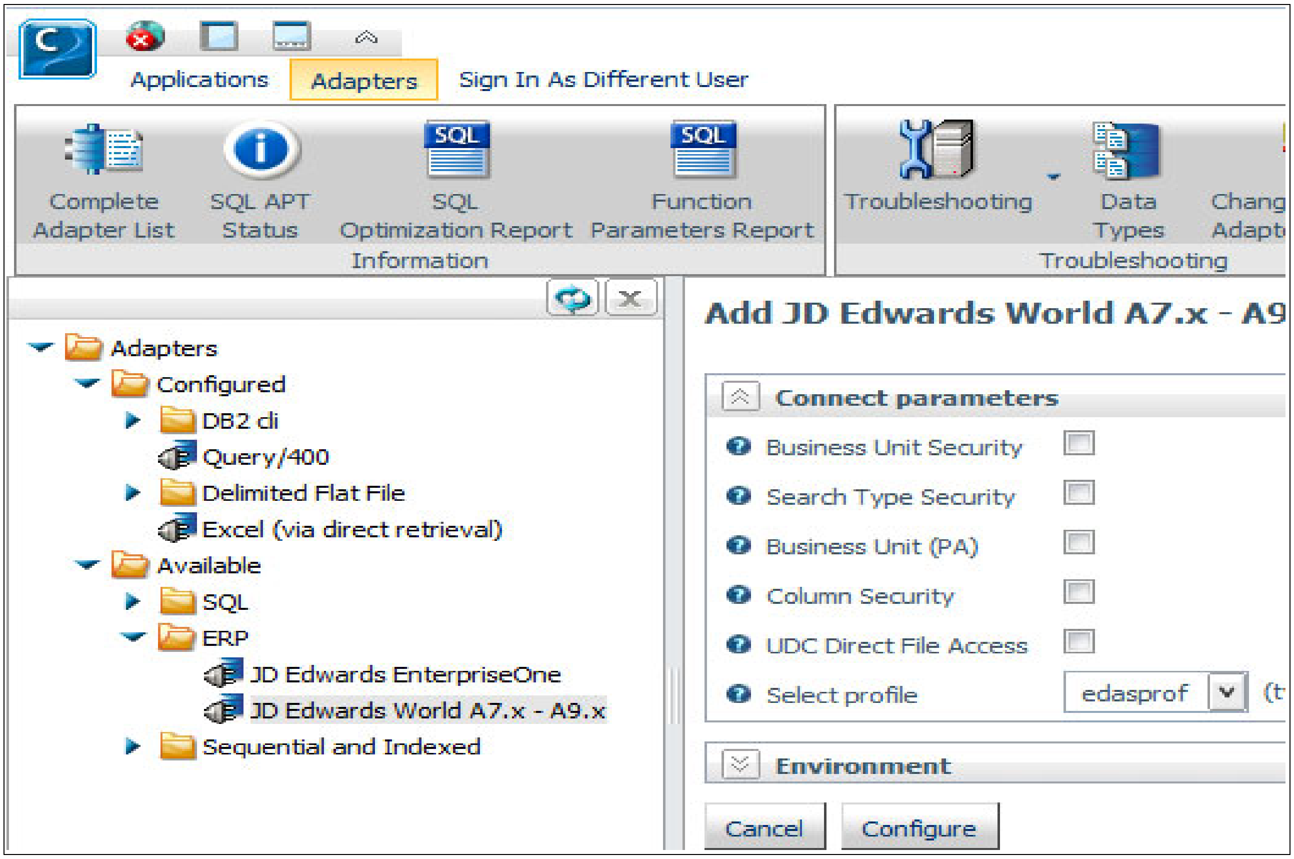
Complete the following steps:

1. Log on to DB2 Web Query by using a user profile that is defined as an Administrator (in the Administrator group within Security Center).

   The administrator is the user that configures and manages the adapter configuration; other users are not permitted to manage and configure adapters.

2. Right-click on a folder in the left panel and select **Metadata** → **Edit**.
3. In the left Adapter navigation pane, expand **Available** → **ERP folder**.

4. Double-click JD Edwards World **A7.x - A9.x**. The configuration window that is shown in Figure 7-9 opens.



Add JD Edwards World A7.x - A9

Connect parameters

Business Unit Security □

Search Type Security □

Business Unit (PA) □

Column Security □

UDC Direct File Access □

Select profile: edasprof (t

Figure 7-9 JD Edward World A7.x -A9.x configuration

5. Select the connection parameters:

   – Business Unit Security

     Select this box to enable automatic execution of JDE World Business Unit Security. Using this automatically restricts user access to data, based on information that is retrieved from the JDE dictionary tables, and then adds appropriate `WHERE` conditions to the user's submitted data access request. Unchecked (OFF) is the default setting. If you select this parameter, you cannot turn it OFF until DB2 Web Query is shut down and then restarted (with no parameter settings).

   – Search Type Security

     Select this box to enable automatic execution of JDE World Search Type Security. The adapter automatically restricts user access to data, based on information that is retrieved from the JDE dictionary table, and then adds the appropriate `WHERE` conditions to the user's submitted data access request. Unchecked (OFF) is the default setting. If you check this parameter, you cannot turn it OFF until DB2 Web Query is shut down and then restarted (with no parameter settings).

   – Business Unit (PA)

     Select this box to revert (if necessary) to an older security model that is used by this adapter. Unchecked (OFF) is the default setting. If checked, this option overrides standard Business Unit Security (as described above).

- Column Security

  Select this box to enable column security based on information in the JDE dictionary file. Unchecked (OFF) is the default setting. If you check this parameter, you cannot turn it OFF until DB2 Web Query is shut down and then restarted (with no parameter settings).

- UDC Direct File Access

  Select this box to enable User-Defined Code Direct File Access. Unchecked (OFF) is the default setting.

> **Important:** This option creates files, `udcdicdb.*`, in the JDWSEC application folder directory. Do *not* delete these files.

a. Select Profile

   This must be EDASPROF.

6. Click **Configure**. You receive a confirmation message. Click **OK**.

Restarting DB2 Web Query disconnects any users, so confirm that no DB2 Web Query user jobs are running before clicking **OK**. After the server restarts, the adapter for JDE World is successfully added to the configuration.

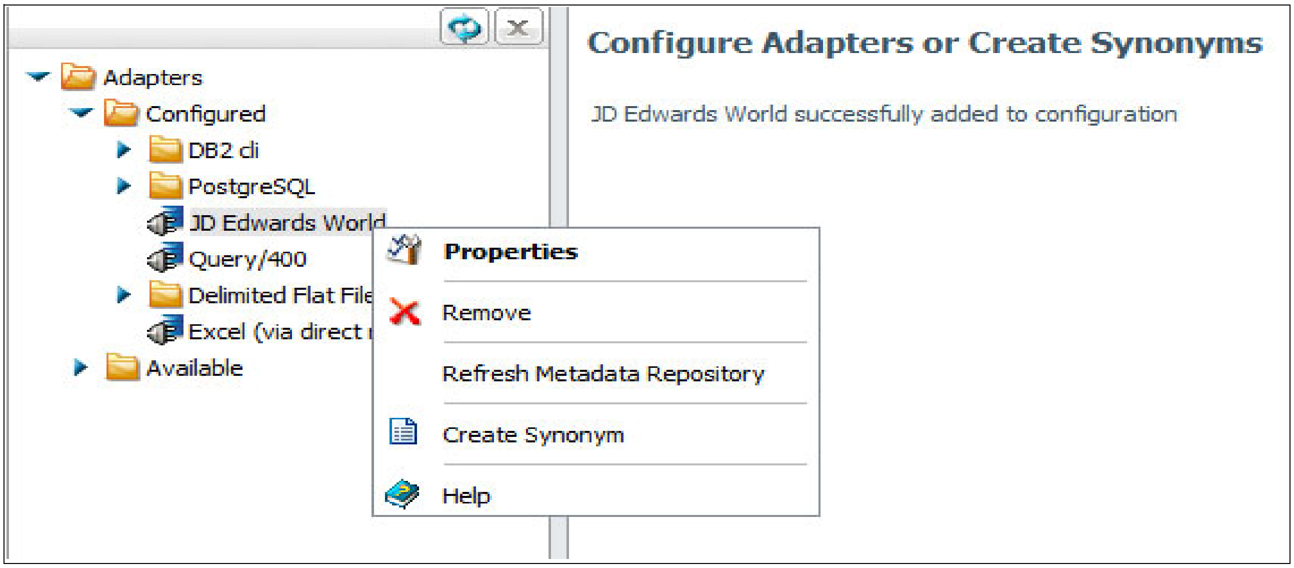
Refreshing the metadata repository

The metadata repository contains the dictionary information for the JDE World files/tables. You must refresh the repository the first time you set up the adapter and repeat the process each time the JDE World tables change.

How to refresh the metadata repository

To refresh metadata, you must first configure the adapter by completing the following steps:

1. Right-click **JD Edwards World** and select **Refresh Metadata Repository**, as shown in Figure 7-10. This step is done only when you first configure the adapter for JD Edwards World, or when JD Edwards data dictionary information changes.



Configure Adapters or Create Synonyms

JD Edwards World successfully added to configuration

Figure 7-10 Menu to refresh the metadata repository

2. Select the version of JDE World that you use. Enter the name of the library for each of the specified objects.

   The UDC library parameter is the library name that contains information about the User-Defined Codes in the JDE dictionary. A new library with the name specified is created on the system. Additionally, a new table is created in that library, which contains UDC information to be used by DB2 Web Query.

3. Click **Refresh Now**.

After the refresh completes, the metadata repository is successfully refreshed.

Creating the JD Edwards World synonyms

To report against JDE World data, you must first create synonyms.

How to create the JD Edwards World synonyms

To create JDE World synonyms, complete the following steps:

1. Right-click **JD Edwards World** and select **Create Synonym**.
2. Click the DB2 CLI connection that points to your JDE World data tables.
3. Select the restrictions that you want to apply when searching for synonym candidates. Restriction options include restrict object type, which further restricts Tables, Views, Aliases, and MQTs.
4. Click **Next**.
5. Select the check box next to the tables for which you want to create synonyms. Click **Create Synonyms**.
6. Add JD Edwards dictionary information to your synonym. The options that you can specify include date format, presumptive join, field names, language code, UDC, and Combine UDC.
   a. Select date format

      The options are YMD, YYMD, DMY, MDY, MDYY, DMYY, MYY, and YYM. (YYMD is the default setting.) The selected format is used only if the field is described as a DATE in the Data Dictionary.

   b. Presumptive Join

      Select the **Presumptive Joins** box to include additional DEFINEs (virtual fields) for presumptive join fields in the synonym. Checked (ON) is the default setting.

   c. Field Names

      Select **Long Fieldname** (the default) to display the field descriptions as names on reports. Select **Short Fieldname** to use the JDE aliases as field names on reports.

   d. Language Code

      Enter the appropriate Language Code, which exists in the JDE F9292 file. (Leave the field blank for English.)

   e. UDC

      Check the UDC box to ensure that UDC description fields are generated as DEFINEs (virtual fields) in the synonym. Selected (ON) is the default setting.

   f. Combine UDC

      Select this box to Combine User-Defined Code. Unchecked (OFF) is the default setting.
7. Click **Continue**.

After you create the synonyms, you can now develop reports to access JD Edwards World data.

7.6.2 Using the adapter for JD Edwards EnterpriseOne

The adapter for JDE EnterpriseOne (E1)allows DB2 Web Query for i to access JD Edwards EnterpriseOne data sources. With this adapter, data in the JDE E1 database is displayed by using rules that are contained in dictionary files, ensuring that valid information is returned to the requesting program.

Preparing the JD Edwards EnterpriseOne environment

Although no environment preparation steps are required, ensure that your system complies with all software specifications.

Do not add this JDE adapter to a DB2 Web Query installation that already contains a configured JDE World adapter. If you want to add this adapter to that server configuration, you must remove the existing JDE World adapter first.

Overview of the setup process

The setup process for the adapter for JDE EnterpriseOne is composed of the following basic steps:

1. Configure the adapter for JDE E1. Define what type of security to implement.
2. Refresh the metadata repository. You must perform this step initially and repeat it only if there are changes in the JDE metadata tables, which occur infrequently at most sites.
3. Refresh the security extract. This step is required only if Group or Role-based security is configured in step 1. This step captures the current JDEdwards E1 rules for the adapter to enforce.
4. Create the JDE E1 synonyms. Synonyms are required for reporting against a data source.

Configuring the adapter for JD Edwards EnterpriseOne

This section explains how to configure the adapter for JD Edwards EnterpriseOne. Complete the following steps:

1. Log on to DB2 Web Query with a user profile of an Administrator (it is defined in Security Center).

   The administrator is the user that configures and manages the adapter configuration; other users are not permitted to manage and configure adapters.
2. Right-click on a folder in the left panel and select **Metadata** → **Edit**.
3. In the new opened window, click the adapter to open the Adapter tab.
4. In the left Adapter navigation pane, expand the **Available** folder.

5. Expand the **ERP** folder. Double-click **JD Edwards EnterpriseOne**; the configuration window opens, as shown in Figure 7-11.

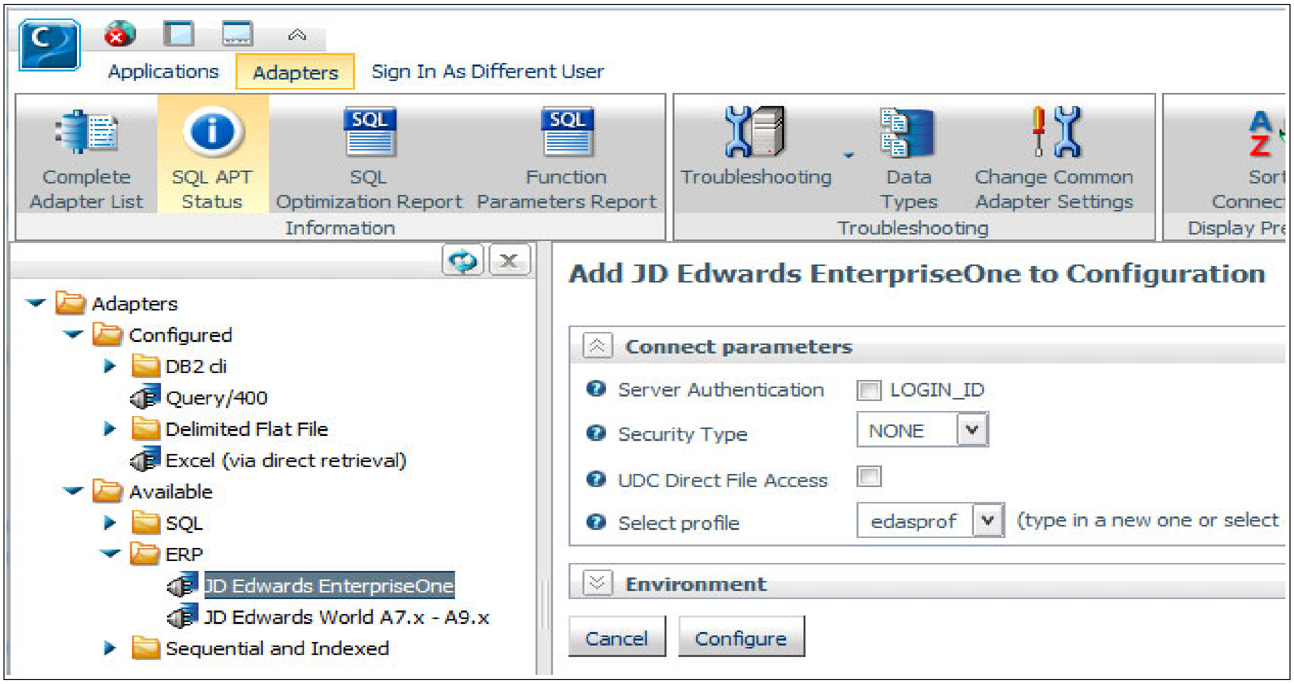


Figure 7-11 JD Edward EnterpriseOne Configuration

6. Select the connection parameters:

   – Server Authentication:

     Select this check box if the reporting server is secure. This option applies when every JDE E1 user has a user ID on the reporting server system, which is the case in DB2 Web Query.

   – Security Type:

     When you configure the adapter for JDE E1, you must choose whether your JDE environment is configured to use role-base security, group-based security, or no security (NONE).

   – UDC Direct File Access:

     When you select this check box, you give users access to the User-Defined Code Direct File.

7. Select **Profile**.

   You must choose edasprof.

8. Click **Configure**.

   You receive a confirmation message.

   > **Important:** The reporting server agents stop. You should confirm that no DB2 Web Query jobs are running before clicking **OK**. Restarting the Reporting Server disconnects any users currently working in DB2 Web Query.

9. Click **OK**.

Refreshing the metadata repository

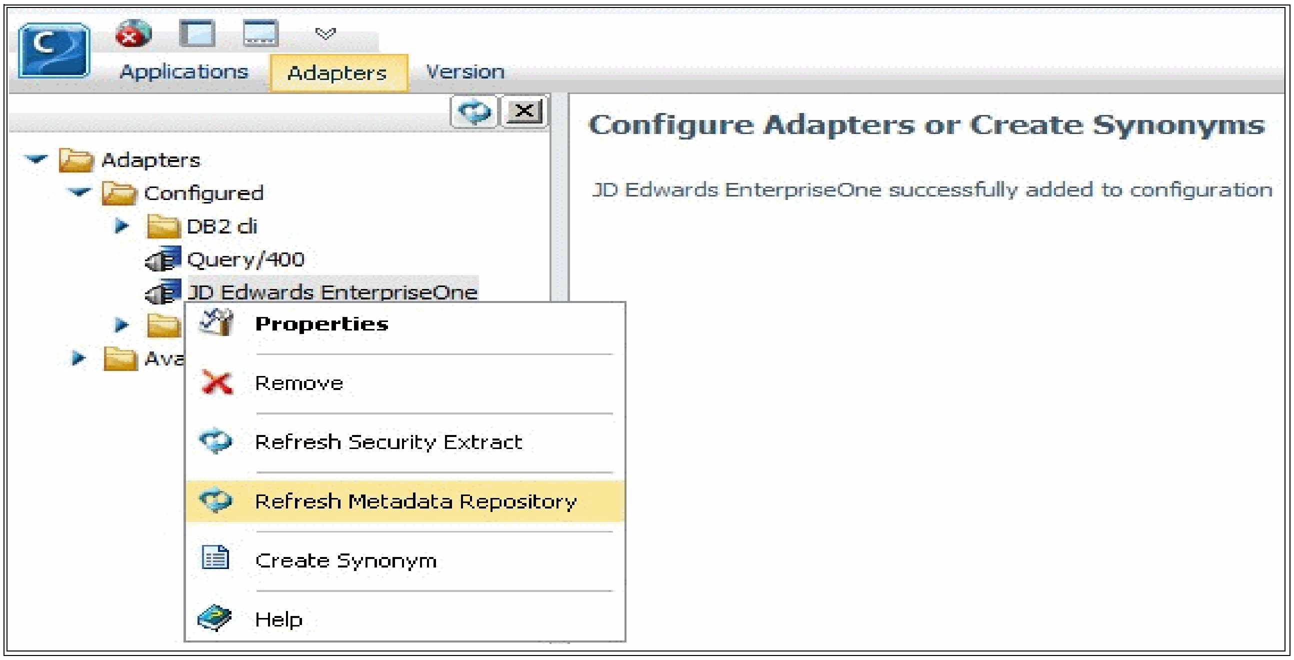
The metadata repository contains the dictionary information for the JD Edwards EnterpriseOne tables.

You must refresh the repository the first time you set up the adapter and repeat the process each time the JDE E1 dictionary tables change.

How to refresh the metadata repository

You can refresh the metadata repository from the Adapters list in the navigation pane on the web page by completing the following steps:

1. Right-click the configured JD Edwards EnterpriseOne adapter and select **Refresh Metadata Repository**, as shown in Figure 7-12.



Configure Adapters or Create Synonyms

JD Edwards EnterpriseOne successfully added to configuration

Figure 7-12 Menu to refresh the metadata repository

   You must perform this step initially, and repeat it only if there are changes in the metadata for tables, which occurs infrequently at most sites.

   The Refresh Metadata Repository pane opens. The JDE tables that are required for this procedure are listed in the first column.

2. Enter the Library name of the library containing the specified objects. The UDC library can be any arbitrary name, for example, UDCDIC.
3. Click **Refresh Now** to refresh the metadata repository.

   After the refresh completes, the metadata repository is successfully refreshed.

Refreshing the security extract

The security extract is a snapshot of the security rules that are defined in JD Edwards EnterpriseOne.

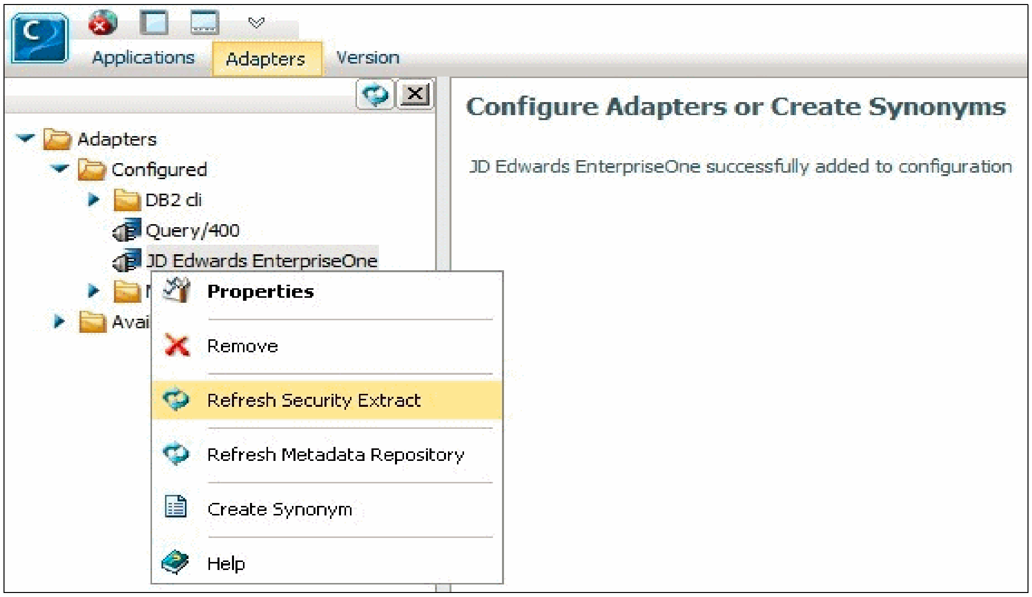
The adapter uses this extract to enforce the restrictions when reporting against JDE E1 data. You must refresh the security extract as often as the security rules change in the application.

How to refresh the security extract

To refresh the security extract, navigate to the Adapters list in the navigation pane on the web page by completing the following steps:

1. Right-click the configured JD Edwards Enterprise One adapter and select **Refresh Security Extract**, as shown in Figure 7-13.

> **Note:** You must perform this step upon initial configuration of the adapter, and repeat it only if there are changes in the JD Edwards EnterpriseOne security rules.



Configure Adapters or Create Synonyms

JD Edwards EnterpriseOne successfully added to configuration

Figure 7-13 Menu to Refresh Security Extract

2. Enter the Library name of the library containing the specified objects.
3. Click **Submit** to refresh the security extract.

Creating the JD Edwards EnterpriseOne synonyms

To report against JDE E1 data, you must first create synonyms.

How to create the JD Edwards EnterpriseOne synonyms

To create the synonyms, complete the following steps:

1. Right-click the JD Edwards EnterpriseOne adapter and select **Create Synonym**.
2. Click the DB2 CLI connection that points to your JDE E1 data tables.
3. Select the restrictions that you want apply when searching for synonym candidates. Restriction options that are included restrict object type, further restricting Tables, Views, Aliases, and MQTs.
4. Click **Next**.

5. Select the parameters that you want the synonym to include. Parameters options include With foreign keys, One-part name, Application, Prefix, Suffix, and Overwrite existing synonyms.

   – With foreign keys

     Select the **With foreign keys** check box to include within this synonym every table that is related to the current table by a foreign key. The resulting multi-table synonym describes all of these tables' foreign key relationships.

   – One-part name

     The One-part name check box is a way to control whether a fully qualified library name is used by the synonym, or whether a library list should always be used. The unchecked behavior generates a table name that includes the explicit name of the library containing the table. For example, if you specified a library on the first Create Synonym pane, a qualified name like the following is automatically created in the Access File:

     ```
     TABLENAME=MYLIB/MYTABLE
     ```

     To avoid hardcoding the library name into any SQL query that is generated by the tool, and instead use library list management, check **One Part Name**. With this type of entry in the Access File, at run time, the library path of the user is searched until the table name is found.

   – Application

     This defaults to the first application folder in the application path.

   – Prefix/Suffix

     You can use prefixes and suffixes to customize the name of the synonym. You might want to do this task to identify the type of synonym, or if you have tables with identical table names, assign a prefix or a suffix to distinguish them. For example, if you have identically named human resources and payroll tables, assign the prefix HR to distinguish the synonyms for the human resources tables. The resulting synonym name cannot exceed 64 characters. If all tables and views have unique names, leave the prefix and suffix fields blank.

   – Overwrite existing synonym

     To specify that this synonym should overwrite any earlier synonym with the same fully qualified name, select the **Overwrite existing synonyms** check box. The connected user must have operating system write privileges to recreate a synonym.

6. Select the check box next to the tables for which you want to create synonyms.
7. Click **Create synonym**.
8. Add JDE dictionary information to the synonym:

   a. Select date format

      The options are YMD, YYMD, DMY, MDY, MDYY, DMYY, MYY, YYM. (YYMD is the default setting.) The selected format is used only if the field is described as a DATE in the Data Dictionary.

   b. UDC

      Check the **UDC** box to ensure that the UDC description fields are generated as DEFINEs (virtual fields) in the synonym. Checked (ON) is the default setting.

   c. Combine UDC

      Check this box to Combine User-Defined Code. Unchecked (OFF) is the default setting.

9. Click **Continue**.

   The synonym is successfully created.

7.6.3 Areas to consider when connecting to a remote JD Edwards database

When connecting to a JD Edwards database, ensure that you account for the following considerations:

1. In the case of the JD Edwards World adapter, the dictionary tables that are replicated get out of sync if the originals are changed.
2. Report data retrieval might be slower than if the tables were local, depending on the amount of data to process.

7.6.4 Using a JD Edwards adapter to connect to a remote JD Edwards database

> **Note:** The JD Edwards dictionary files are needed on the local system to access JD Edwards databases. See the following sections for names of these files.

The JD Edwards World adapter and the JD EnterpriseOne adapter function differently in how they access the dictionary information:

- The JD Edwards World adapter does not use an extract of the dictionary files for the Master File conversion step, but does have a UDC descriptions extract locally to speed up the UDC lookup when running a report.
- The JD Edwards EnterpriseOne adapter uses an extract for all of the dictionary information.

The metadata repository for the adapter must be re-created if there are changes to the information in UDC or security tables, which apply to both adapters, but principally the JD Edwards EnterpriseOne adapter. If the repository is not refreshed, then information in the report might be incorrect, or access to the data might be restricted or allowed depending on the security change.

7.6.5 JD Edwards dictionary files that are needed on local system to access remote JD Edwards databases

If you are connecting to a remote LPAR/Server where the JD Edwards applications are running on a different LPAR or server where DB2 Web Query is installed, the following synonym files must be copied for JD Edwards to use the database:

- F0004
- F0005
- F0006
- F0010
- F0101JB
- F0013
- F9202
- F9201

These files are needed for JD Edwards World 8 and higher:

- F0004

- F0005
- F0006
- F0010
- F0101JB
- F0013
- F9202
- F9210

These files are needed for JD Edwards EnterpriseOne:

- F0004
- F0005
- F0013
- F9202
- F9210

These files must be copied to the local system in a library that is in the DB2 Web Query for i startup library list and the user profile library list. That library name also is used in the entries that are needed when configuring the JD Edwards adapters.

8

Managing data with DataMigrator

As mentioned in Chapter 1, “The cycle of Business Intelligence and IBM DB2 Web Query for i” on page 1, it is often advantageous to use an operational data store (ODS) or data warehouse (DW) to prepare or “untangle” data for reporting. Although an ODS or DW is often the preferred foundation for good reporting and analytics, populating and maintaining such an environment can often be seen as a large or mundane task. This perception can frequently prevent the creation of such an environment, resulting in diminished reporting effectiveness and likely overburdening an already busy business production environment.

To the rescue comes a great tool for the job: DataMigrator!

To address the challenges with creating and maintaining an ODS or DW, IBM provides an Extract, Transform, and Load (ETL) tool called DataMigrator.

DataMigrator is the shortened name for the newest member of the IBM DB2 Web Query for i product family. It provides an ETL solution for building, populating, and maintaining database tables from one or more data sources. Therefore, it is a great fit for untangling data.

Like all DB2 Web Query products, DataMigrator runs directly on IBM i. It is built for any IBM i customer interested in consolidating data for reporting and analytics, or for creating a more optimal reporting environment that does not interfere with an existing production environment.

DataMigrator is available as a companion product for either the Express or Standard edition of DB2 Web Query. Because it is a companion product, either Express (5733-WQE) or Standard (5733-WQS) is required as a prerequisite product before ordering DataMigrator.

This chapter provides overview information about DataMigrator and describes some features of what can be done with it.

8.1 Product overview

DataMigrator provides the following capabilities:

- Reads source data from a host of different data sources, including database tables (files), flat (IFS) files, IBM i journals, and even procedures that return result sets.
- Populates a set of target tables, such as a data warehouse partition.
- The ability to apply transformation logic to sanitize and convert data into a cleansed, optimized form.
- Supports aggregating data to simplify reporting and decision support.
- Automates all data movement, either through its built-in scheduler or by running data flows from your favorite IBM i job scheduler (the latter requires Standard Edition).

8.2 Product components

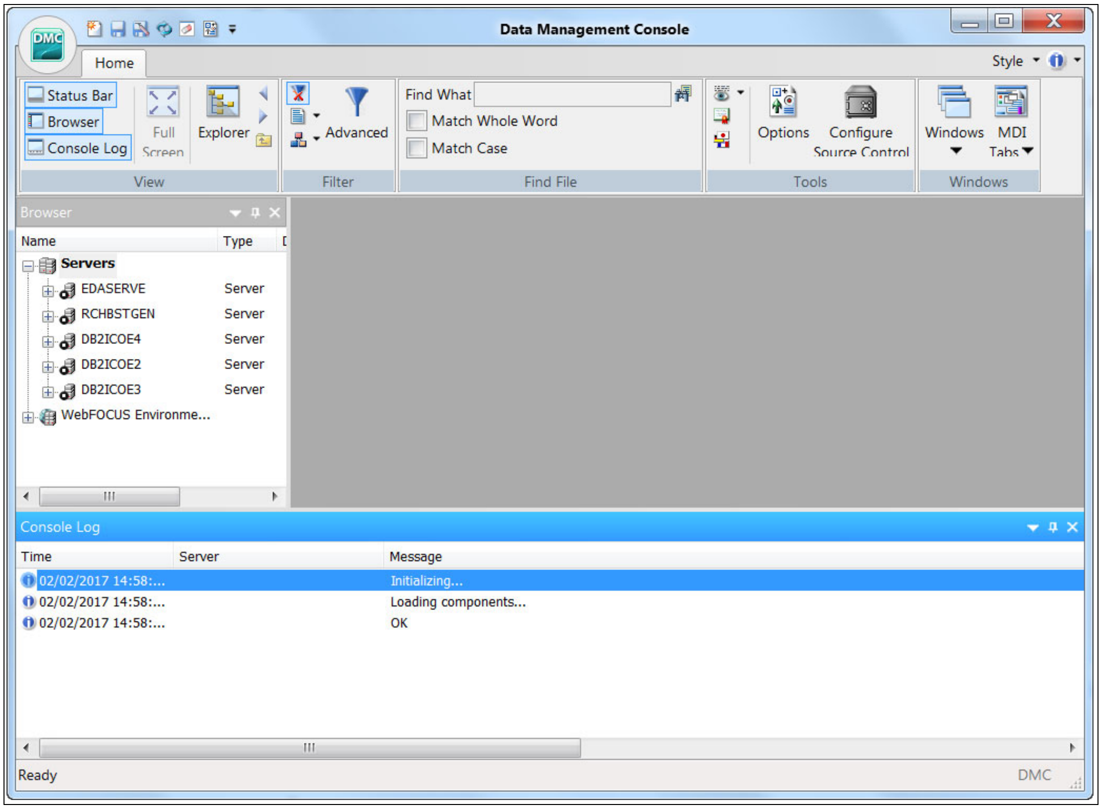
DataMigrator consists of the following major product components described in this section.

8.2.1 Data Management Console

The Data Management Console (DMC) is part of the DB2 Web Query Developer Workbench development toolkit and shares the same license. This windows based installation is used to define data and process flows. Data sources, targets, and transformations are defined within data flows through a series of drag and drop operations. The DMC interface visualizes data flows to show how data transforms from the source to the target.

DMC is also used for defining and scheduling process flows, defining email notification of flow outcomes, viewing run logs, and reviewing data statistics.

The DMC has a similar interface to Developer Workbench. Its main window is shown in Figure 8-1.



Name	Type
Servers	
EDASERVE	Server
RCHBSTGEN	Server
DB2ICOE4	Server
DB2ICOE2	Server
DB2ICOE3	Server
WebFOCUS Environme...	

Time	Server	Message
02/02/2017 14:58:...		Initializing...
02/02/2017 14:58:...		Loading components...
02/02/2017 14:58:...		OK

Figure 8-1 Data Management Console

Note: Although the DMC is used to build flows and define the schedule for those flows, it is not involved in the actual running of the flows themselves. Flows are stored and run directly on IBM i.

8.2.2 The DataMigrator server

The DataMigrator Server is the back-end component of DataMigrator that stores and runs data flows. It also processes defined schedules. This server runs on the IBM i and runs in the same instance (jobs) as the DB2 Web Query server. Figure 8-2 shows the architecture of DataMigrator server.

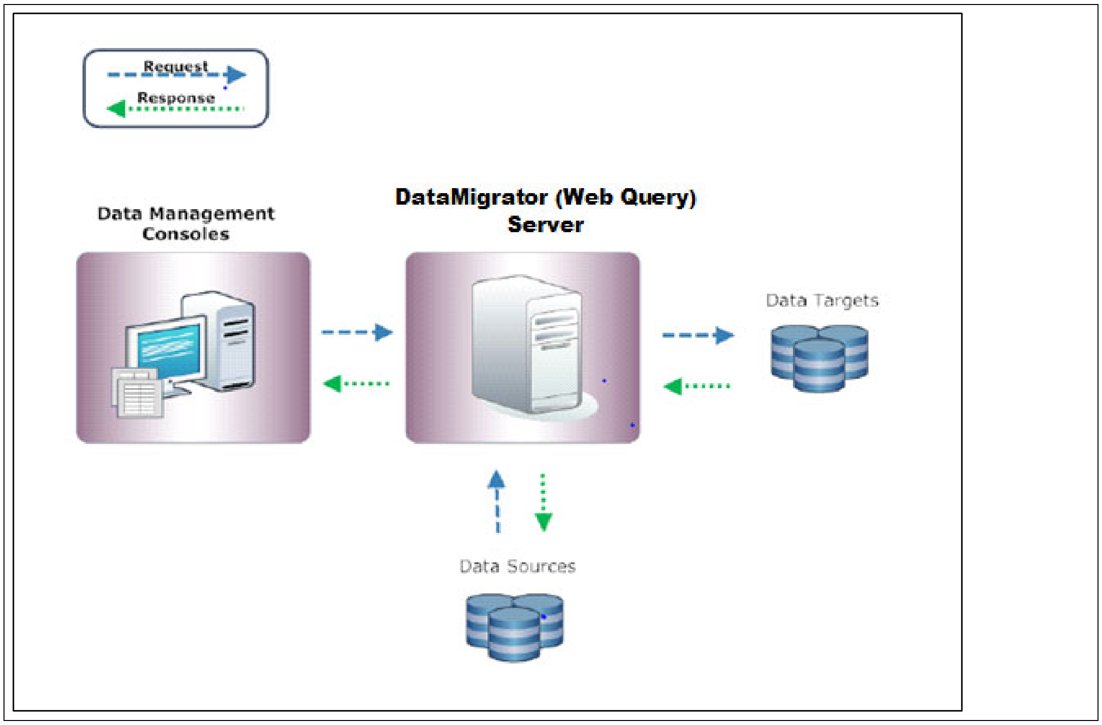


Figure 8-2 DataMigrator server

8.3 Product installation

Although DataMigrator is ordered as a separate ordering product ID (PID) 5733-WQM, the installation of the product itself is integrated within the DB2 Web Query for i product 5733-WQX as option 8.

Using the same media that you used for the original DB2 Web Query for i installation, install OPTION(8) on the RSTLICPGM. Additional, specify **`FEATURE(5108)`** with the **`ADDLICKEY`** command to register the license keys for DataMigrator for i.

Note that DB2 Web Query EZ-Install includes, and automatically does the RSTLICPGM for option 8 noted above. Like all the products/features shipped with EZ-Install, you can run DataMigrator for 70 days under a trial period before license keys would be required.

8.4 Product examples: Flows

Flows within DataMigrator come in two major forms: data flows and process flows.

Data flows define the actual steps for reading, transforming and writing data.

Process flows organize the order and way in which one or more data flows run.

Each time a new flow is created in DataMigrator, both a data flow and process flow are created and connected together. The process flow is automatically populated with one step and that step is to run the data flow. DMC then immediately positions the user within the data flow palate so that the work of defining the data movement can begin.

Within DataMigrator, many options are available for replicating and transforming data. Data flows define those transformations, the source and target databases.

What follows are a few examples of DataMigrator data flows. There are many options for data sources; these are only examples of the most common ones. Note that with DB2 Web Query Standard Edition, even non-IBM i based data can be used as data sources in an ETL architecture.

This first Data Flow example shows a simple flow from a table from library **QWQCENT** to an output table, as shown in Figure 8-3. However, even in this simple case considerable data transformation can take place.

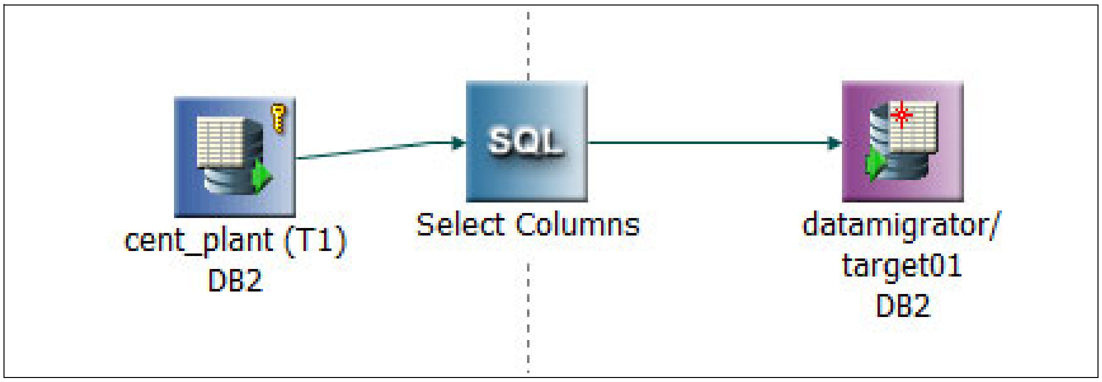


Figure 8-3 Simple flow

For the next Data Flow example, there are two inputs: two tables joined together. This example also shows the output going to multiple target tables. This multi-target output is useful in populating, for example, multiple dimension tables in a DW at once or creating workfiles that are used to hold intermediate steps in the process. Figure 8-4 on page 178 shows the data flow.

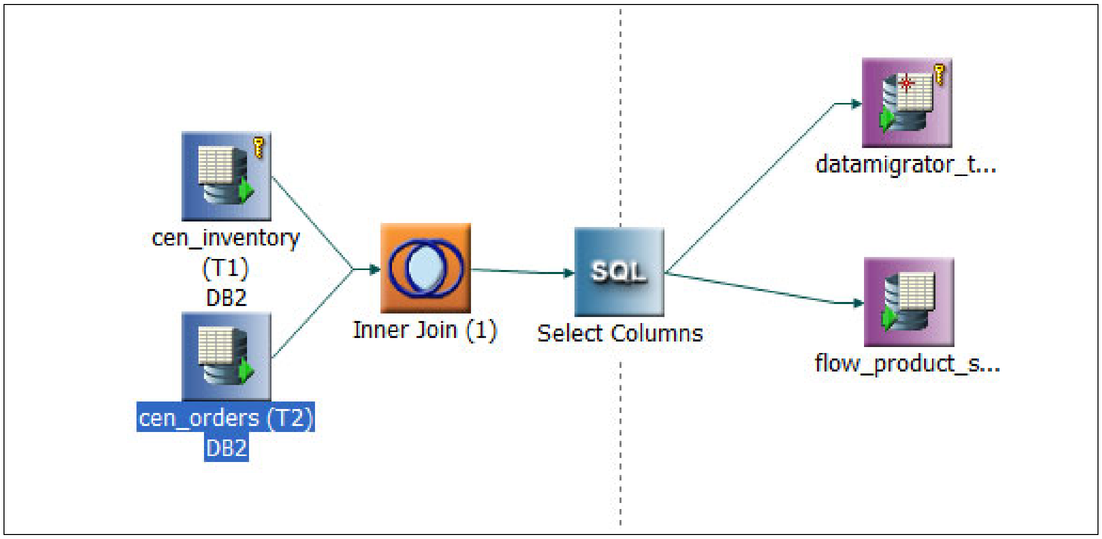


Figure 8-4 Data flow with a join

One of the main uses cases for DataMigrator is to capture only changed data from the source database and update the target. This is called *change data capture* (CDC) and can provide a huge performance benefit over reprocess all data from the beginning each time an update is needed. The most common way to do CDC is to read journals that are associated with source tables, transform the data as necessary, and write to the target.

To accommodate practical CDC scenarios, DataMigrator keeps track of its journal position using the associated synonym of the journal. In this way, as data flows run using the journal synonym, the position within the journal is kept up to date to ensure each subsequent run of the data flow continues from the tracked position.

This example shows a simple case of reading from a journal, mapping the data as necessary, and writing to a target table in the DW, as shown in Figure 8-5.

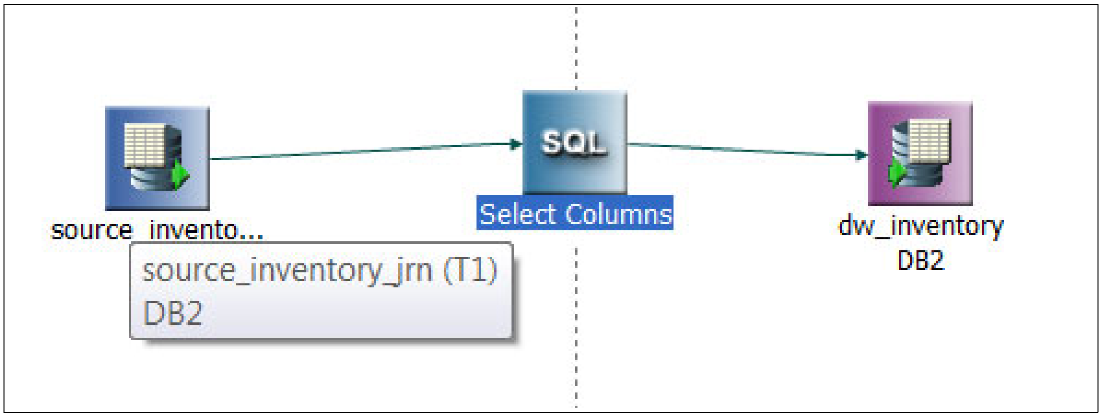


Figure 8-5 Data flow with a join

Note: In Figure 8-5, the source journal is a remote journal. In addition to local journals, remote journals are also supported as a data source by DataMigrator. Remote journals are a great, low system impact way to move raw data to the target system in order for DataMigrator to consume it.

The next example expands on the previous journal example. It shows a Data Flow that has two inputs: a remote journal and a table joined together. This is a technique that is employed often, as the raw data being read from the journal often needs to be transformed into the warehouse layout by first looking up in a table such as a dimension table to get the final values. An example of this is for a star schema based data warehouse. In that case, the incoming rows from the journal are cross referenced into a dimension table, which then produces the dimension table's key, which is finally put into the fact table, as shown in Figure 8-6.

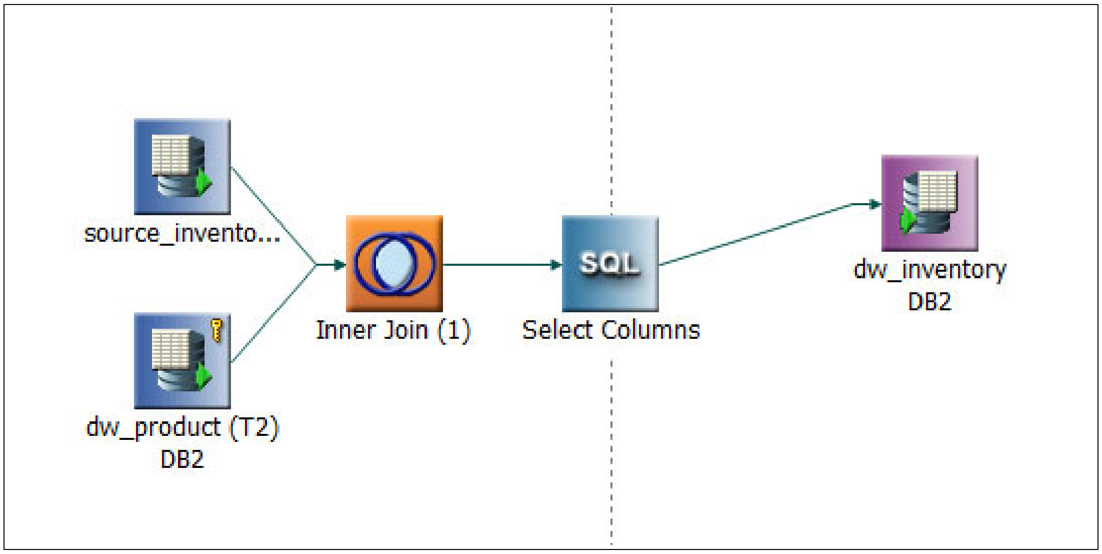


Figure 8-6 Data flow of two journals

As mentioned earlier, a process flow is used to control an ETL job's overall flow. Each data flow has a default process flow automatically created for it. In general it is not necessary for the user to even know about the process flow.

However, building more involved process flows can be a powerful way to control more complex operations. A good example of a process flow's importance is to set up a stream of data flows where one data flow runs only after the previous flow has completed successfully.

This final example leverages a process flow. See Figure 8-7. This process flow calls one data flow, then a second data flow after the first one completes, then sends an email when both data flows have completed.

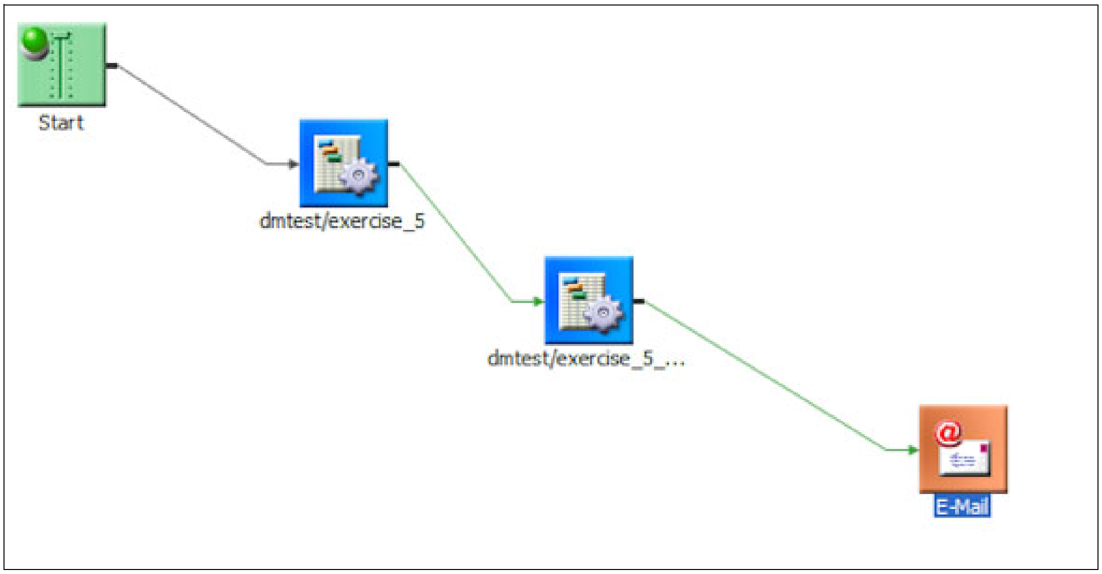


Figure 8-7 Process flow

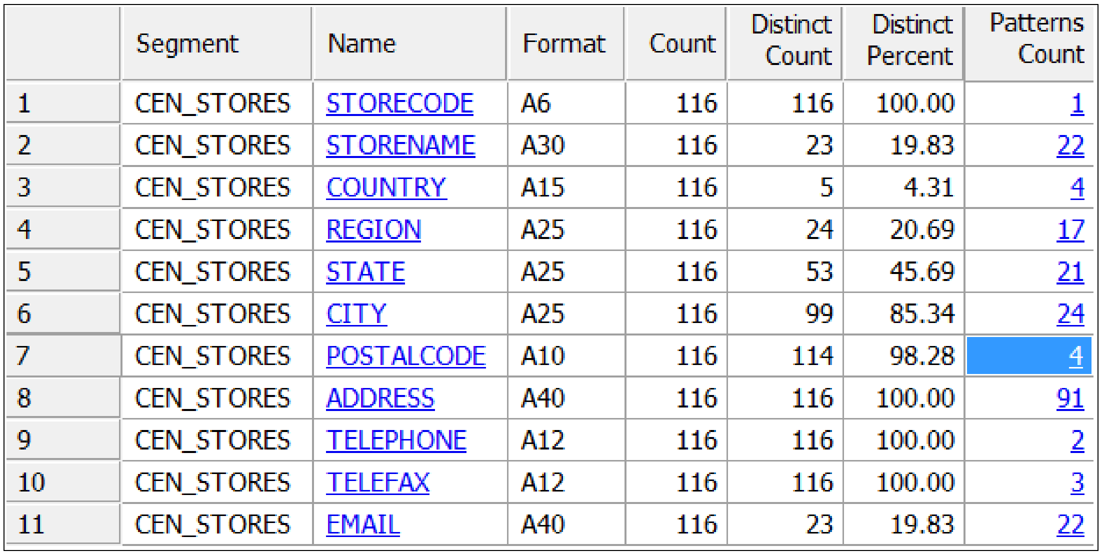
8.5 Product examples: Data profiling

Data profiling is a DataMigrator feature that provides the ability to "examine" data in a data source, at a high level, to better understand the data format and patterns in the data.

The following examples illustrate the data profiling capability.

Note: These examples use the sample database table STORES in library QWQCENT, which is shipped and installed with DB2 Web Query for i.

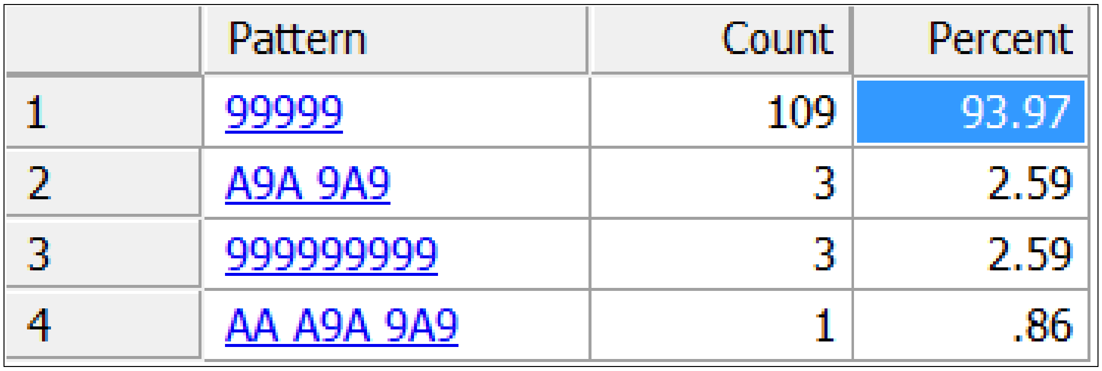
This first example has two screen shots and shows the patterns that are found within the POSTAL CODE in the STORES table. Figure 8-8 shows that POSTALCODE has four patterns.



	Segment	Name	Format	Count	Distinct Count	Distinct Percent	Patterns Count
1	CEN_STORES	STORECODE	A6	116	116	100.00	1
2	CEN_STORES	STORENAME	A30	116	23	19.83	22
3	CEN_STORES	COUNTRY	A15	116	5	4.31	4
4	CEN_STORES	REGION	A25	116	24	20.69	17
5	CEN_STORES	STATE	A25	116	53	45.69	21
6	CEN_STORES	CITY	A25	116	99	85.34	24
7	CEN_STORES	POSTALCODE	A10	116	114	98.28	4
8	CEN_STORES	ADDRESS	A40	116	116	100.00	91
9	CEN_STORES	TELEPHONE	A12	116	116	100.00	2
10	CEN_STORES	TELEFAX	A12	116	116	100.00	3
11	CEN_STORES	EMAIL	A40	116	23	19.83	22

Figure 8-8 Patterns with postal code

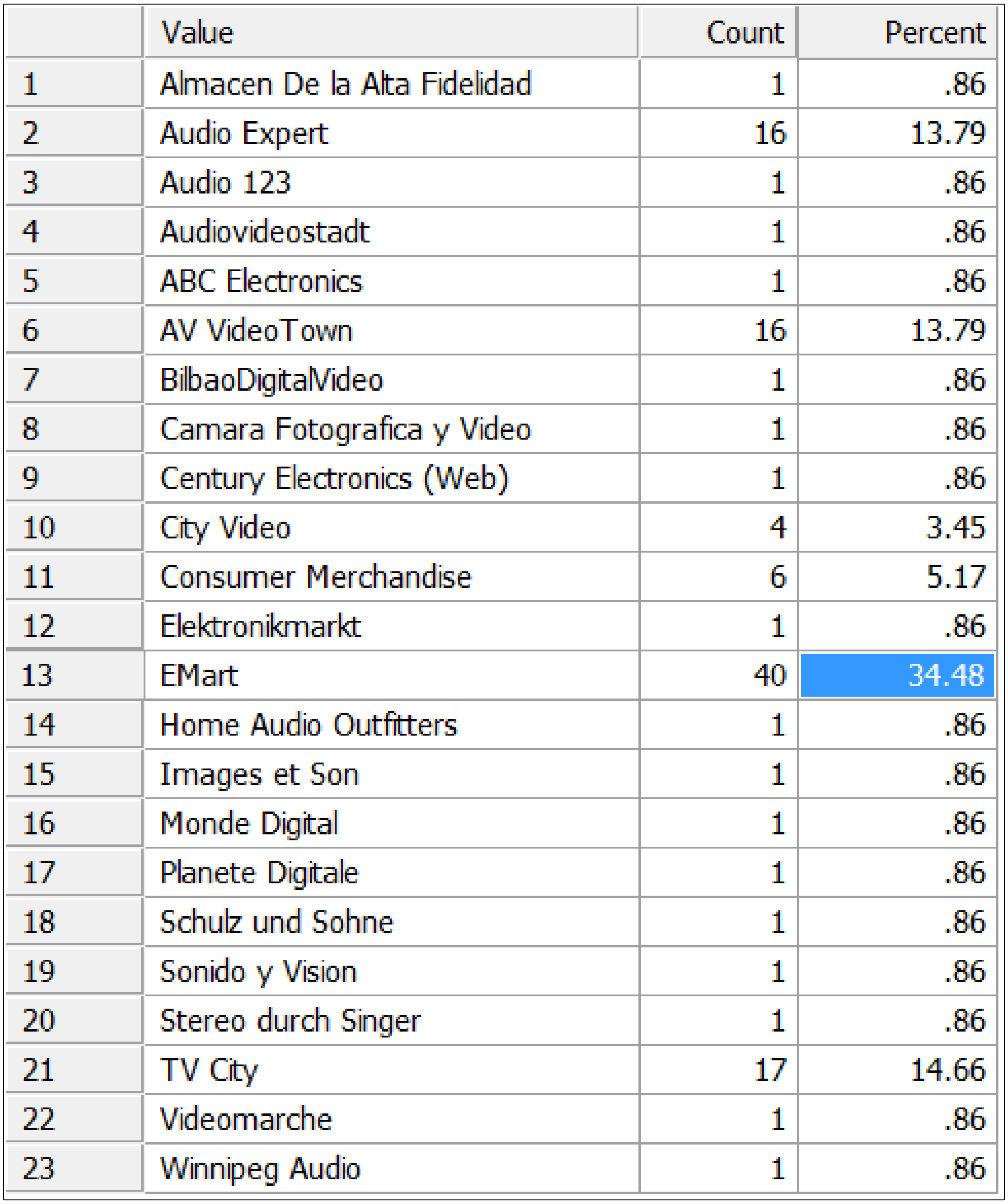
Figure 8-9 shows the pattern types and their distribution.



	Pattern	Count	Percent
1	99999	109	93.97
2	A9A 9A9	3	2.59
3	999999999	3	2.59
4	AA A9A 9A9	1	.86

Figure 8-9 Distribution of patterns

This next data profiling example highlights a case of data skew. Data skew occurs when a single (or small number) value is prevalent in the data much more than any other value. Continuing the analysis on the STORES table, this time we focus on the STORENAME column and its actual values. Figure 8-10 shows that there is a high percentage of one store name in the table.



	Value	Count	Percent
1	Almacen De la Alta Fidelidad	1	.86
2	Audio Expert	16	13.79
3	Audio 123	1	.86
4	Audiovideostadt	1	.86
5	ABC Electronics	1	.86
6	AV VideoTown	16	13.79
7	BilbaoDigitalVideo	1	.86
8	Camara Fotografica y Video	1	.86
9	Century Electronics (Web)	1	.86
10	City Video	4	3.45
11	Consumer Merchandise	6	5.17
12	Elektronikmarkt	1	.86
13	EMart	40	34.48
14	Home Audio Outfitters	1	.86
15	Images et Son	1	.86
16	Monde Digital	1	.86
17	Planete Digitale	1	.86
18	Schulz und Sohne	1	.86
19	Sonido y Vision	1	.86
20	Stereo durch Singer	1	.86
21	TV City	17	14.66
22	Videomarche	1	.86
23	Winnipeg Audio	1	.86

Figure 8-10 Data profiling over store name

8.6 Summary and more information

This chapter provided a short introduction to DataMigrator to illustrate the power of the product. For more information about the product's capabilities and to get started with using the product to untangle your data, refer to the DataMigrator User's Guide.

Integrating web data into your custom application

Today, most companies have their own website (or portal), and it is common to want to integrate static information with dynamic information, such as publishing up-to-date business and performance indicator or simply giving each registered user the ability to link into information of their specific interest. You can use DB2 Web Query to fulfill this need.

The DB2 Web Query REST-based Application Extension (WQRAX) is a capability built into DB2 Web Query Standard Edition feature. WQRAX allows a reporting object to be run and the output presented by invoking a URL string from the application. This allows web browsers or other web applications to run and display those reports directly from the application. WQRAX eliminates the need for users to log in to the DB2 Web Query BI Portal, find a report, and run it.

The reporting object type can be any of the following:

- Report
- Graph
- Dashboard
- Compound document
- HTML page

The basic premise of WQRAX is simple: Each reporting object is represented by a specific URL. To run the report from a web application, simply invoke the representative URL. Detailed instructions can be found in the *REST-based Application Extension Administration Guide*, which can be downloaded from the DB2 Web Query wiki documentation link.

To run DB2 Web Query reports through WQRAX, a registered licensed user or a member of a registered runtime group must be used for validation.

9.1 Initial setup

WQRAX does not require a specific installation. A default configuration is provided under Standard Edition. Therefore, it is available for use immediately.

Several configuration options do exist to control specific behaviors of WQRAX, which are stored in a configurable properties file. An interface is provided to alter them. For more information about the properties file and the various options, see 9.7, “Properties file and options” on page 193.

9.2 Using the extension

WQRAX is an abstraction layer that is based on REST-based web services within the product. Its intention is to simplify the use and integration of the web services, but it does require a basic understanding of the modes of use and the various ways to authenticate the user. These topics are described in this section.

9.2.1 Modes of use

WQRAX has two modes of use: browse and direct. Each is explained in the following sections.

Browse mode

All reporting object content is physically stored in the DB2 Web Query repository. However, from a logical, user interface perspective, these objects are organized and stored in either top-level folders or subfolders in a tree-like structure. Users can find these objects by “traversing the tree” and navigating to the folder that contains the objects.

The browse mode of WQRAX provides an interface for users to traverse the tree and browse the reporting object content without knowing the specific URL for a top-level folder, subfolder, or report. This mode displays the contents of the folder to users, allowing them to browse the contents of the folder. They can then simply click a report to run it or they can click a subfolder to jump to that level and show the subfolder contents (which can be reports or more subfolders). As with the BI Portal interface, only content that the user is authorized to see is shown.

You must have browse mode enabled in the properties file. For more information, see 9.7, “Properties file and options” on page 193.

To begin working in browse mode, simply open the following URL in your web browser:

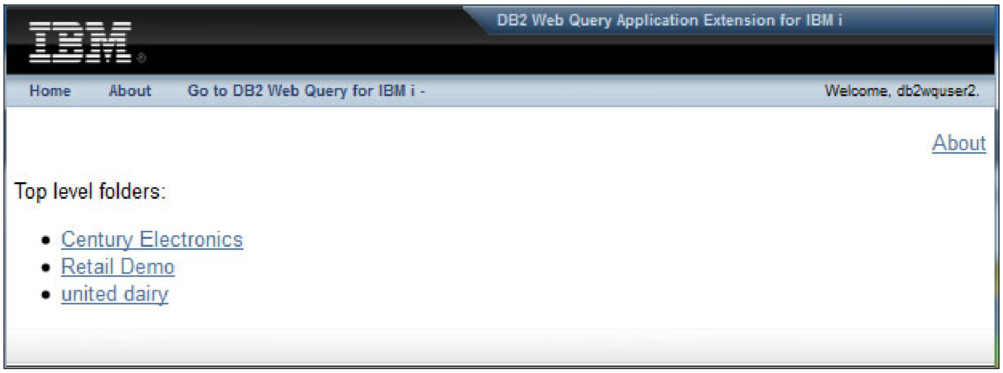
```
http://<yourserver>:12331/wqrax
```

Where *<yourserver>* is your IBM IP address or DNS name.

> **Note:** This link is similar to one that is used to log in to the BI Portal:
>
> ```
> http://<yourserver>:12331/webquery
> ```
>
> The same port number (12331) is used for both; only the last portion is different.

With the installation default configuration, you are prompted to log in with your IBM i user profile and password. You must be either a registered licensed DB2 Web Query user or a member of a registered runtime group. For more information about authentication, see 9.4, “Authentication methods” on page 189.

The security implementation is consistent with the BI Portal interface. After you are logged in, all the top-level folders that you are authorized to see are shown, as shown in Figure 9-1.


IBM
DB2 Web Query Application Extension for IBM i
Home
About
Go to DB2 Web Query for IBM i -
Welcome, db2wquser2.
About

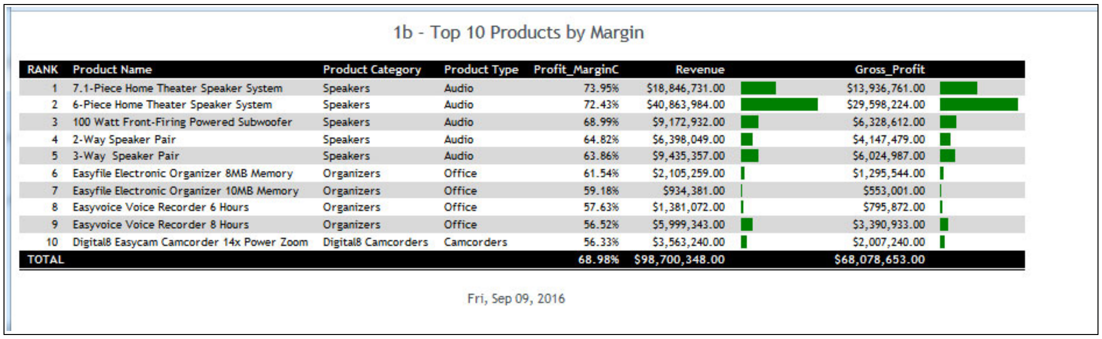
Top level folders:

- Century Electronics
- Retail Demo
- united dairy

Figure 9-1 WQRAX top-level folder

Simply click a top-level folder to see a list of its contents. These contents might be subfolders or reporting objects. Selecting an object runs it.

When a report runs, the results are shown in the browser (see Figure 9-2) if the report output is HTML or Active Reports; for other formats, they are handled according to the format specification.



1b - Top 10 Products by Margin

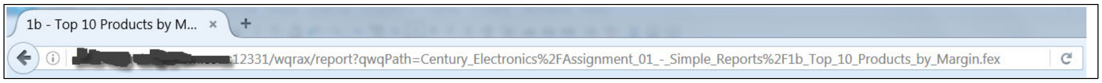
RANK	Product Name	Product Category	Product Type	Profit_MarginC	Revenue	Gross_Profit
1	7.1-Piece Home Theater Speaker System	Speakers	Audio	73.95%	$18,846,731.00	$13,936,761.00
2	6-Piece Home Theater Speaker System	Speakers	Audio	72.43%	$40,863,984.00	$29,598,224.00
3	100 Watt Front-Firing Powered Subwoofer	Speakers	Audio	68.99%	$9,172,932.00	$6,328,612.00
4	2-Way Speaker Pair	Speakers	Audio	64.82%	$6,398,049.00	$4,147,479.00
5	3-Way Speaker Pair	Speakers	Audio	63.86%	$9,435,357.00	$6,024,987.00
6	Easyfile Electronic Organizer 8MB Memory	Organizers	Office	61.54%	$2,105,259.00	$1,295,544.00
7	Easyfile Electronic Organizer 10MB Memory	Organizers	Office	59.18%	$934,381.00	$553,001.00
8	Easyvoice Voice Recorder 6 Hours	Organizers	Office	57.63%	$1,381,072.00	$795,872.00
9	Easyvoice Voice Recorder 8 Hours	Organizers	Office	56.52%	$5,999,343.00	$3,390,933.00
10	Digital8 Easycam Camcorder 14x Power Zoom	Digital8 Camcorders	Camcorders	56.33%	$3,563,240.00	$2,007,240.00
TOTAL				68.98%	$98,700,348.00	$68,078,653.00

Fri, Sep 09, 2016

Figure 9-2 WQRAX report that is run

The user is prompted to enter any necessary parameter values if there are filters that are set in the report.

At any time during the tree navigation process, the URL bar contains the top-level folder, subfolder, and report, as shown in Figure 9-3. This is a key concept when using the direct mode method.



[redacted]12331/wqrax/report?qwqPath=Century_Electronics%2FAssignment_01_-_Simple_Reports%2F1b_Top_10_Products_by_Margin.fex

Figure 9-3 WQRAX URL

9.2.2 Direct mode

When the specific URL of the report to be run is known, that URL can be invoked directly, which eliminates the need for users to traverse the tree in browse mode and allows them to run the report directly from an application.

The structure of direct mode is consistent with that of browse mode; the only difference is that the “full path” of the report is specified as the value of the qwqPath construct in the URL. For example, to run the report with the full path value of `My_Top_Level_Folder/My_Report.fex` on your server, specify the following URL:

```
http://your_server:12331/wqrax/report?qwqPath=My_Top_Level_Folder/My_Report.fex
```

After you construct the URL, that string can be used directly in various implementations. For example:

- Link to it from a website or PC application.
- Set it as the source of an HTML iframe (in-line frame) on an existing web page.
- Email the link to your colleagues.

Because it is a URL, every user must have access to the IBM i and be a valid DB2 Web Query user or member of a registered runtime group. In a corporate environment, this might mean establishing a VPN connection or gaining clearance to a firewall that might be protecting your server.

The URL is static and persistent, which means that if the report still exists in the same folder structure and the user has authority to it, it can always be used to run the report. The URL can be bookmarked, copied/pasted into another web application, or sent as a link in an email.

9.3 Report parameter considerations

Many reports have filters to allow users to select a subset of the data at run time. When those reports run, users are prompted for parameters so that they can specify the value for those filters.

When such reports are invoked by using WQRAX, the parameters can either be specified in the URL (specifying a predefined set of data) or they can be omitted entirely from the URL. If omitted, WQRAX automatically builds the necessary prompt controls and prompts the user to specify/select the missing values, which is the case for both browse and direct mode.

Here is an example of calling a report with parameters that are preset in the URL:

```
http://your_server:12331/wqrax/report?qwqPath=My_Top_Level_Folder%2FMy_Report.fex&
COUNTRY=United States&PRODUCTTYPE=Office
```

9.4 Authentication methods

As is the case when running reports in the i BI Portal, users of WQRAX must identify themselves to DB2 Web Query through an authentication process.

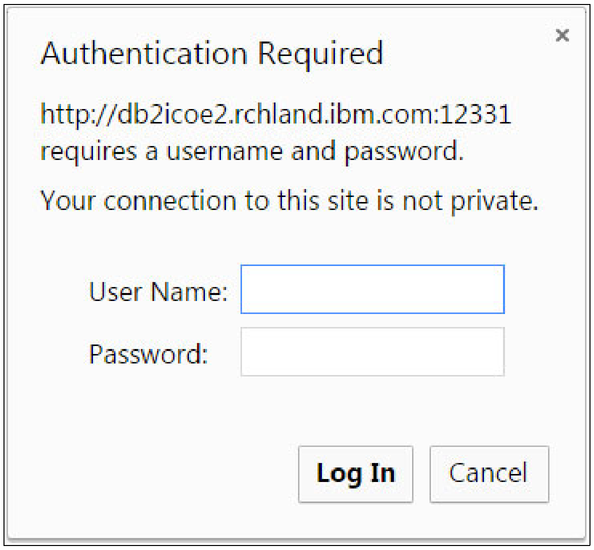
WQRAX provides four distinct methods of authentication:

- Basic authentication
- URL parameter authentication
- Static authentication
- Application specified authentication

If a single sign-on (SSO) environment is enabled and DB2 Web Query is configured to participate in it, WQRAX acknowledges it, which means that SSO enabled users no longer receive an authentication prompt by the web browser. Instead, Windows kerberos credentials are used for authentication. Instructions to configure DB2 Web Query with SSO (SPNEGO web authentication) are available from the DB2 Web Query wiki.

9.4.1 Basic authentication

Basic authentication means that users are prompted for their IBM i credentials the first time that they attempt to use WQRAX in the current browser session. An example prompt is shown in Figure 9-4.



Authentication Required

http://db2icoe2.rchland.ibm.com:12331 requires a username and password.

Your connection to this site is not private.

Figure 9-4 WQRAX: Basic authentication logon prompt

The user provides their IBM i user profile and password. The user profile must be either a registered licensed user or a member of a registered runtime group.

Once authenticated, the necessary information is encrypted and passed along in the http header of subsequent WQRAX requests in the same browser session. This avoids users being prompted for their credentials for each WQRAX request; it is done once per session. If you close your browser and reopen it, you are prompted to enter your credentials again.

Basic authentication is the default (shipped) configuration. It is enabled by specifying the following snippet in the WQRAX properties file:

```
wqraxBasicAuthEnabled=true
```

Note: When basic authentication is enabled, it is always enforced and users are always prompted for their credentials. If you want to use another method of authentication, you must first disable basic authentication.

9.4.2 URL parameter authentication

The user profile and password values can be specified as parameters in the URL when a report is requested through WQRAX. The parameter names are `&wqraxUser` and `&wqraxPassword` (the ampersand prefix denotes they are parameters and is required). For example, to run a report and log in as user profile FRED with password ABC123, the following URL is specified:

```
http://your_server:12331/wqrax/report?qwqPath=My_Top_Level_Folder%2FMy_Report.fex&
wqraxUser=FRED&wqraxPassword=ABC123
```

Basic authentication must be disabled for this method of authentication to be used. Specify the following snippet in the properties file:

```
wqraxBasicAuthEnabled=false
```

Note: With this authentication method, the password is readable text.

9.4.3 Static authentication

If you are in a corporate environment where DB2 Web Query is accessed from within a secure intranet configuration and you have no need to distinguish between individual users of WQRAX, you can effectively bypass the login process.

This method is accomplished by having each user log in implicitly by using a shared IBM i user profile. The shared user profile must also be either a registered licensed DB2 Web Query user or a member of a registered runtime group. The user profile and password that is used to log in all users is stored in the properties file. Here is an example properties file configuration:

```
user=FRED
password=ABC123
```

In this example, all users of WQRAX are logged in as user FRED; there is no other form of authentication that is performed.

Note: Use with caution. All users look the same to the DB2 Web Query for i application.

If you are using reports that depend on knowing the true identity of the user, for example, reports based on SQL views with row level filter security, this method is not recommended.

Basic authentication must be disabled for this method of authentication to be used. Specify the following snippet in the properties file:

```
wqraxBasicAuthEnabled=false
```

9.4.4 Application authentication

A fourth method of authenticating users is by specifying the credentials from within a web-based application before the first WQRAX request is submitted. In this situation, there is a web application that is controlling or directing access into DB2 Web Query and the intent is to integrate reports as part of that overall application.

Basic authentication must be disabled for this method of authentication to be accepted. Specify the following snippet in the properties file:

```
wqraxBasicAuthEnabled=false
```

In this mode, the application becomes responsible for performing the authentication in DB2 Web Query for i on behalf of the user. The application passes the authentication information through a POST method (for security). This process requires consideration of two aspects:

1. The application is driving the authentication.
2. The application must work with the browser to get the session authentication set in the browser on behalf of the user.

When the application must authenticate to DB2 Web Query (usually the first time the user navigates to the applicable part of the application), the application sends a POST request that includes the **`wqraxUser`** and **`wqraxPassword`** parameters.

The POST returns a token, which is a simple string. The getToken request is acknowledged on a POST request only for increased security. Attempting to invoke it directly from the browser's URL (a GET request) fails to return a token.

You can use AJAX to run the signOn request asynchronously. This approach might be preferable to make the page experience more seamless, and faster, for the user.

In either case, when the JavaScript/form is run, it performs the actual sign-on authentication under the browser's control. The browser automatically handles the resulting authentication session ID, freeing the application from having to deal with the authentication session ID during the session.

After the authentication is complete, the user's session is authenticated to WQRAX.

Now, the application can show additional WQRAX URLs on the web page for the user to click, for example, `http://<yourserver>:12331/wqrax` for browse mode or `http://<yourserver>:12331/wqrax/report?qwqPath=top_level_folder/report.fex` for direct mode.

In addition, if a report contains drilldowns or launches new pages, WQRAX automatically passes the authentication information for those reports.

Optional WQRAX_USER parameter

In some environments, you might want to use a shared user profile to authenticate to DB2 Web Query for multiple users, but still provide some way to uniquely identify each user within the reports.

WQRAX provides this capability through the reserved **WQRAX_USER** parameter. **WQRAX_USER** is a predefined parameter within the WQRAX environment that can be optionally set and used. It is referenced as a global variable in a DB2 Web Query synonym and its value is passed “silently” throughout the WQRAX environment. It provides a way for reports to filter on a “user”, even though a shared user profile is used.

To use it, the application must set **WQRAX_USER** at the same time as when the authentication token is being requested. WQRAX then associates the given **WQRAX_USER** value as the parameter’s value for the duration of the session.

> **Note:** To use the **WQRAX_USER** parameter, a one line fex must be created in the Common folder and accessible to all WQRAX users. For more information, see the **WQRAX_USER** fex parameter in the properties file (9.7.16, “WQRAX_USER fex” on page 196).
>
> **Note:** if a report references **WQRAX_USER** as a parameter, but the application fails to set **WQRAX_USER** during the authentication, the user is prompted to enter **WQRAX_USER**. This is unwanted, but it provides an easy way to debug when an application fails to set the **WQRAX_USER** value correctly.

9.5 Dynamic Run Time Environment support

If Dynamic Run Time Environments (RTEs) is used to set a library list for metadata resolution, it can be used with WQRAX too. WQRAX supports RTE settings and acknowledges the currently active RTE for the logged-in user. Therefore, if a library list or an exit program is configured for the user’s active RTE, the library list is changed and the exit program is called before the report is run through WQRAX.

In addition, the active RTE can be dynamically changed at any time in WQRAX by using the **qwqActiveRuntimeEnvironment** parameter.

Specifying **qwqActiveRuntimeEnvironment=***your_runtime_environment_name* in the URL as a parameter changes the current active runtime environment to *your_runtime_environment_name*.

Any subsequent request to run reports later on continue to acknowledge the active runtime environment you just set.

This is an example of changing the active RTE in WQRAX while running a report (*Gross_Profit_Active_Report* is the name of the report, *QWQCENT* is the RTE name):

```
http://my_server:12331/wqrax/report?qwqPath=RTEtest%2FGross_Profit_Active_Report.f
ex&qwqActiveRuntimeEnvironment=QWQCENT
```

9.6 Restrictions

WQRAX has some restrictions that must be considered during report design and implementation.

- No InfoMini support

  The WQRAX extension does not support reports that generate InfoMini output. Any reports that are started from WQRAX that include InfoMini output do not render or function correctly.

- No OLAP support

  The WQRAX extension does not support reports that use the OLAP feature. Any reports that are started from WQRAX that include OLAP output do not render or function correctly.

9.7 Properties file and options

WQRAX behavior is ruled by a properties file that is stored in the IFS:

```
/qibm/userdata/qwebqry/extensions/wqrax.properties
```

By default, the properties file is set to provide functions without any additional configuration. However, there are properties that you can change with a few simple parameters. This section describes each setting in the properties file. Each section has a brief comment that is included in a line that is prefaced by a hash (#).

DB2 Web Query provides a web interface at `http://server:12331/wqrax/config` to make changes to this file. You must have *SECADM special authority in addition to being an administrator in DB2 Web Query to access this page.

> **Note:** DB2 Web Query must be restarted for changes to the properties file to become operational.

9.7.1 Server

This parameter defines the DB2 Web Query server to which to connect. By default, it is localhost, that is, the local system. However, it is possible to have more than one server with DB2 Web Query on it and you might want to point the application extension running on one server to DB2 Web Query running on the other. DB2 Web Query still checks the license on the system where the application extension is running, even if it points to a remote DB2 Web Query server. Here is an example value:

```
server=localhost
```

> **Note:** A valid license is required on both local and remote systems when a different server than LOCAL is set in this parameter.

9.7.2 Port

This parameter is the port that DB2 Web Query is running on. This should always be 12331. Here is an example value:

```
port=12331
```

9.7.3 Basic authentication

This parameter determines the WQRAX login credentials approach for each user. By default, the setting is `true`, which means that the user is prompted with an authentication window. Used with the user and password parameters or properties, this parameter can be set to `false` to use alternate methods for authentication, as described in 9.4, “Authentication methods” on page 189. Here is an example value:

```
wqraxBasicAuthEnabled=true
```

9.7.4 User and password

These parameters define the shared user profile that is commonly used by a set of users when using the static authentication method that provided by `wqraxBasicAuthEnabled=false`. Here are example values:

```
user=MYUSRPRF
password=myPassword
```

9.7.5 Browse mode

This parameter determines whether browse mode is enabled. By default, is it set to `true`. For more information, see 9.2.1, “Modes of use” on page 186. For security reasons, you might choose to disable browse mode by setting this parameter to `false` and allow only direct access mode. Specify `true` to enable and `false` to disable this feature. Here is an example value:

```
browse=true
```

9.7.6 Cache timeout

This parameter determines the number of seconds before any cached report graphs are discarded. Here is an example value:

```
cacheTimeout=14400
```

9.7.7 Token timeout

This parameter determines the number of seconds before the token that is returned from the getToken method becomes invalid. Here is an example value:

```
tokenTimeout=7200
```

9.7.8 Template heading

This parameter controls what is used as the header for each HTML page. Here is an example value:

```
wqraxTemplateHead=ibm_header.html
```

9.7.9 Template footing

This parameter controls what is used as the footer for each HTML page. Here is an example value:

```
wqraxTemplateFoot=ibm_footer.html
```

9.7.10 Parameter prompt limit

This parameter controls the maximum number of parameters that WQRAX prompts for on a given report. Here is an example value:

```
wqraxSelectListSize=5
```

9.7.11 Result schema

This parameter controls the target schema (library) for storing saved reports and statistics. Here is an example value:

```
wqraxSchema=QWQRAX
```

9.7.12 Enable statistics

This parameter controls whether statistics about WQRAX run reports are kept. Here is an example value:

```
wqraxEnableStats=false
```

9.7.13 Debug mode

This parameter determines whether debug messages are sent to the servlet log. You might need to add `com.ibm.ejs.ras.level=INFO` into the servlet's tracing specification if it is not already there. Use this option only under guidance from a support representative. Here is an example value:

```
debug=false
```

9.7.14 Parameter prompt

This parameter determines whether the user is prompted for any parameters that are not provided with the URL invocation. By default, the user is prompted. Here is an example value:

```
wqraxPrompt=true
```

9.7.15 About link

This parameter determines whether the 'About' link shows for the main browser session. Here is an example value:

```
wqraxShowAboutLink=true
```

9.7.16 WQRAX_USER fex

This parameter determines the fex (report) that should be run to set the **WQRAX_USER** global variable. By default, it is set rax_user.fex. Here is an example value:

```
wqraxUserFex=rax_user.fex
```

Note: The target fex (rax_user.fex) must be:

- In the DB2 Web Query Common folder
- Publicly accessible for WQRAX users
- A one line fex with the following statement:

  ```
  -SET &&WQRAX_USER = &INUSER;
  ```

The fex itself is *not* shipped with the product. It *must* be created by you if you intend to use **WQRAX_USER**.

9.7.17 Timeout redirect URL

WQRAX invokes the DB2 Web Query REST web service APIs to run reports. Before running the report, WQRAX must connect to REST to sign on. The connections to REST can time out, according to the session timeout configuration.

It is possible to configure a WQRAX timeout redirect URL in wqrax.properties. When the connection to REST times out, WQRAX redirects the user to the configured URL. Here is an example value:

```
wqraxRedirectURL=http://www.google.com
```

Note: This action happens only in environments where the **wqraxUserFex** parameter is set up; if so, the users are redirected to the configured URL.

When the **wqraxUserFex** parameter is not set, WQRAX automatically reconnects to REST if the connection times out.

Only users who set up the **wqraxUserFex** parameter are redirected to the configured URL; for other users, WQRAX automatically reconnects to REST if the connection times out.

9.7.18 Email output

This parameter enables sending the output report through email. It applies only to reports with parameters. By default, it is set to false to not allow sending. Here is an example value:

```
wqraxEnableBackgroundReports=false
```

9.7.19 SMTP server

This parameter is used to set up the email server to be used when the sending of background report is enabled (`wqraxEnableBackgroundReports=true`). The server should be a valid SMTP mail server without credentials. The server does not need to be running on your IBM i system. Here is an example value:

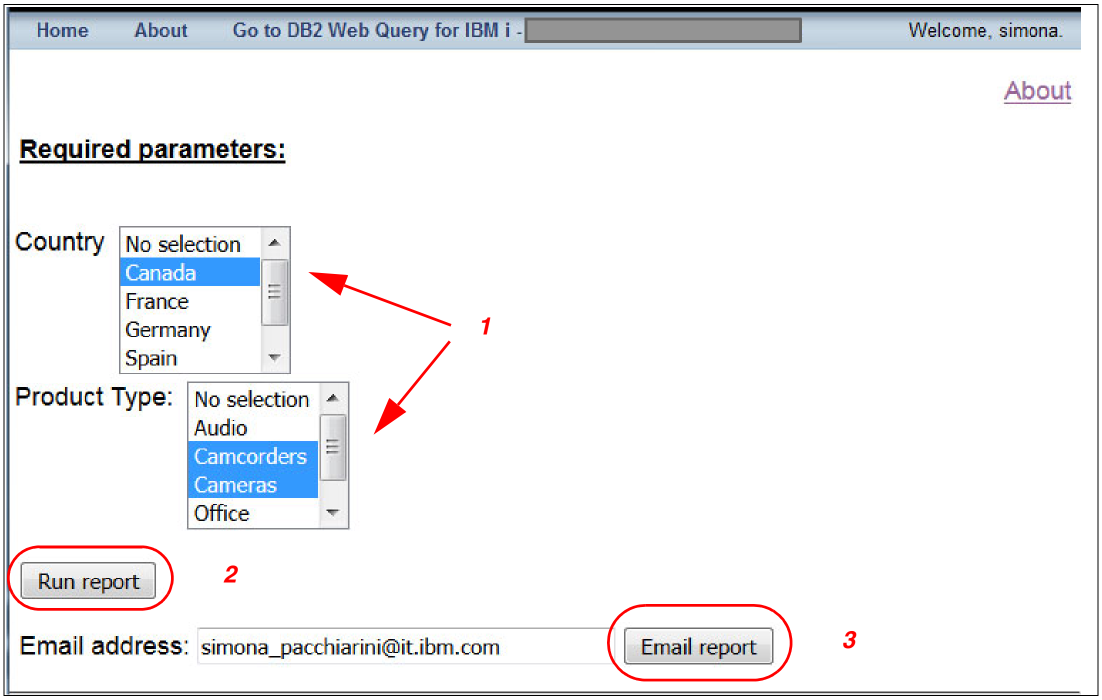
```
wqraxSMTPServer=localhost
```

9.7.20 Sender

With this parameter, you set up the sender for any beggared report that is sent through WQRAX. This parameter is considered only when `wqraxEnableBackgroundReports=true`. Here is an example value:

```
wqraxSMTPSender="WQRAX Report Service" <no-reply@localhost>
```

Figure 9-5 shows an example of using WQRAX to run a report with a parameter when sending through email is enabled. After setting the parameters (filters) to the wanted values 1, you can decide whether you simply want to run the report 2 or if you want to send the output by email 3.



Required parameters:

Country: No selection, **Canada** (selected), France, Germany, Spain

Product Type: No selection, Audio, **Camcorders** (selected), **Cameras** (selected), Office

Email address: simona_pacchiarini@it.ibm.com

Figure 9-5 WQRAX report with parameters and email output

A

DB2 Web Query amper variables

Amper variables are DB2 Web Query variables whose names are prefixed with a single ampersand (&). An example is `&DATE`, which indicates the current date. Unlike other variables or fields whose values are determined by the underlying data source, the values for amper variables are resolved at run time. However, like other variables, they can be used in metadata or for display. There are two categories of amper variables: global and local.

Global amper variables are also referred to as *system variables*. They are available globally for use by all reports, and they are predefined and automatically supplied by the system when a procedure references them. `&DATE` is a system variable.

Local amper variables are referred to as *dynamic prompting variables* or *auto-prompt variables*. They are optional. They are local to a report or its metadata and are defined by the developer or DBA. Values for the variables are provided by a user running the report or by a user scheduling the report.

There are unlimited uses for amper variables. One simple and common usage is to display them in a header or footer. More powerful capabilities can be used by integrating them into define fields, filters, or other metadata operations.

System variables

System variables are predefined in DB2 Web Query and are automatically resolved when a procedure (report) runs. They are available when running the procedure in the BI portal, through Report Broker schedules, from Developer Workbench, and from the WQRAX application extension.

Table A-1 lists the system variables available in DB2 Web Query.

Table A-1 System variables

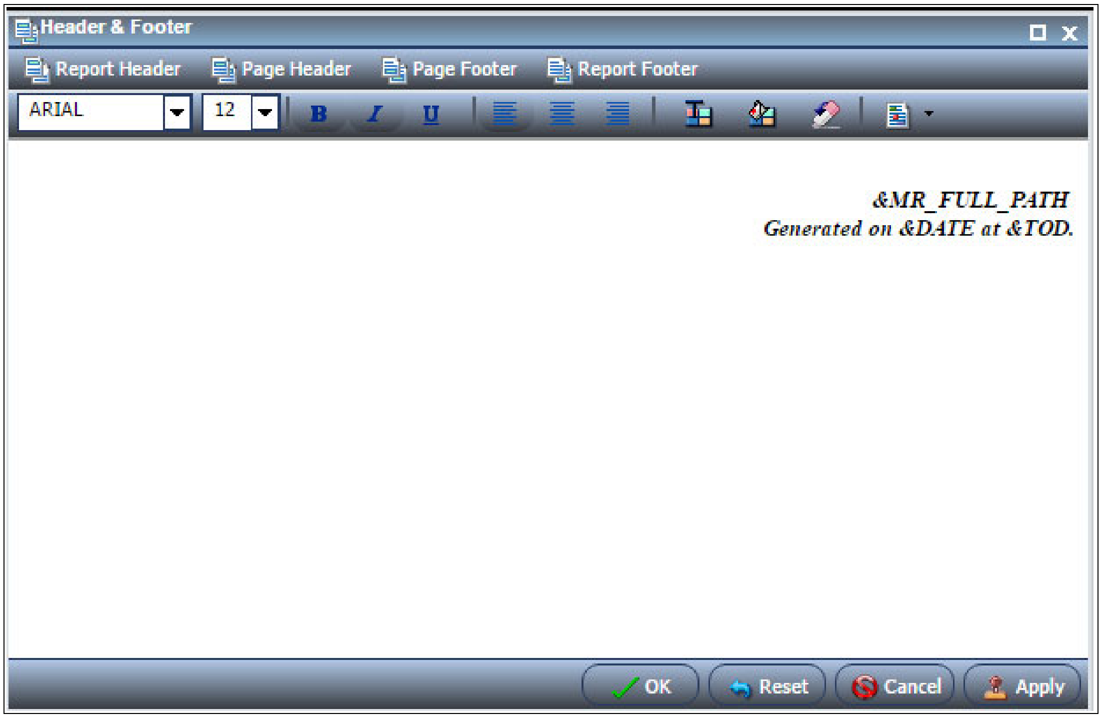
Variable	Format or value	Description
&DATE	MM/DD/YY	Returns the current date.
&DATEWtr	Name of day of week	Returns the full name of the day of the week (for example, Wednesday).
&DATEMtrDYY	Name of month DD, YYYY	Returns the name of the month followed by the day and the four-digit year (example, October 7, 2016).
&DATEWtr, &DATEMtrDYY	Name of day of week, name of month DD, YYYY	Returns the full name of the day of the week, followed by the name of the month, followed by the day and the four-digit year (example, Wednesday, October 7, 2016).
&DMY	DDMMYY	Returns the current date.
&DMYY	DDMMCCYY	Returns the current (four-digit year) date.
&FOCCODEPAGE	Any integer value	Returns the code page being used by the server.
&FOCFEXNAME		Returns the name of the FOCEXEC that is running. This variable differs from the &FOCFOCEXEC variable because &FOCFOCEXEC returns the name of the calling FOCEXEC only.
&FOCFIELDNAME	NEW OLD NOTRUNC	Returns a string indicating whether long and qualified field names are supported. A value of OLD means that they are not supported, NEW means that they are supported, and NOTRUNC means that they are supported, but unique truncations of field names cannot be used.
&FOCFOCEXEC		Returns the fully qualified path for the procedure. This variable is deprecated for DB2 Web Query. Use &FOCFEXNAME instead.
&FOCMODE	AS400	Identifies the operating environment.
&FOCNEXTPAGE	0	Variable whose value is determined by the last page number that is used by the last report. Its value is one more than the last page number that is used in the last report.
&FOCQUALCHAR	. : ! % | \	Returns the character that is used to separate the components of the qualified field names.
&FOCREL	Release number	Identifies the FOCUS Release number (for example, 6.5 or 6.8).

Variable	Format or value	Description
&FOCSBORDER	ON OFF	Whether solid borders are used in full-screen mode.
&FOCUSER		Returns the connected user ID. Similar to the **GETUSER** function.
&MR_FULL_FEXNAME		Returns the full FEX name. This is the same name that appears in the DB2 Web Query Managed Reporting (MR) interface from the web browser.
&MR_FULL_PATH		Returns the complete path for a procedure, including the folder where the report is stored, the file name, and extension. (This is the same value as shown in the BI portal when right-clicking a report and selecting **Show Path**.) An example full path is `IBFS:/WFC/Repository/Century_Electronics/test/myreport.fex`.
&MDY	MMDDYY	Returns the current date.
&MDYY	MMDDCCYY	Returns the current (four-digit year) date.
&TOD	HH.MM.SS	Returns the current time.
&YMD	YYMMDD	Returns the current date.
&YYMD	CCYYMMDD	Returns the current (four-digit year) date.

Note: Because many date format options can be appended to the prefix *DATE* to form one of these variable names, avoid using DATE as the prefix when creating your own amper variable names.

Here is an example use case for system variables. If you have hundreds of reports, and if a user (such as an executive or sales manager) requests a modification to one of them, you must know which one it is. The advantage of including the full path variable in a header or footer is that if allows easy identification of the report. Including the date and time variables provides an additional reference point for when the report was generated.

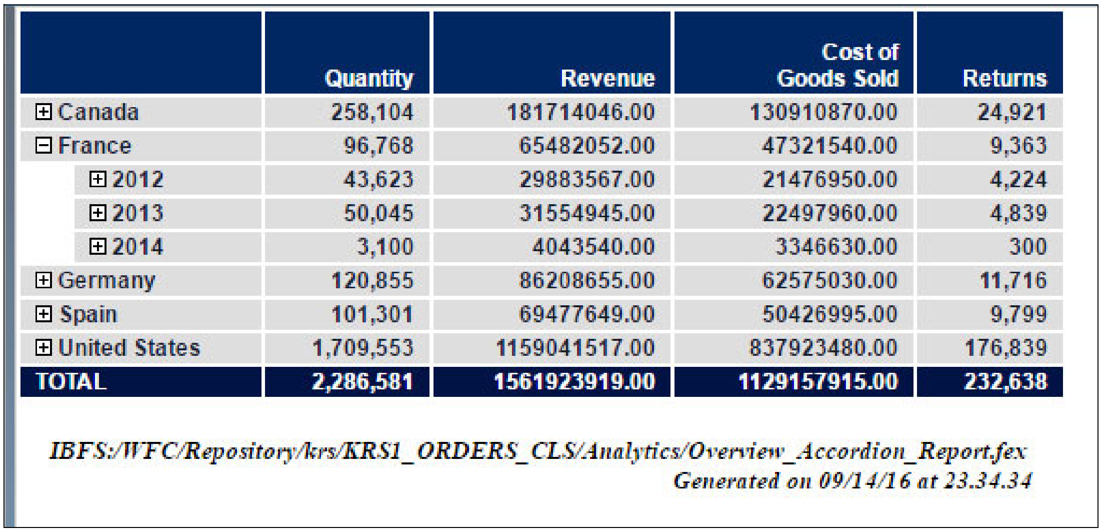
Figure A-1 shows an example of the full path, date, and time variables that are edited into a report footer.



&MR_FULL_PATH
Generated on &DATE at &TOD.

Figure A-1 Full path, date, and time variables

Figure A-2 shows the report output.



	Quantity	Revenue	Cost of Goods Sold	Returns
⊞ Canada	258,104	181714046.00	130910870.00	24,921
⊟ France	96,768	65482052.00	47321540.00	9,363
⊞ 2012	43,623	29883567.00	21476950.00	4,224
⊞ 2013	50,045	31554945.00	22497960.00	4,839
⊞ 2014	3,100	4043540.00	3346630.00	300
⊞ Germany	120,855	86208655.00	62575030.00	11,716
⊞ Spain	101,301	69477649.00	50426995.00	9,799
⊞ United States	1,709,553	1159041517.00	837923480.00	176,839
TOTAL	2,286,581	1561923919.00	1129157915.00	232,638

IBFS:/WFC/Repository/krs/KRS1_ORDERS_CLS/Analytics/Overview_Accordion_Report.fex
Generated on 09/14/16 at 23.34.34

Figure A-2 Accordion report with a customized footer

You can use amper variables in headers and footers for more sophisticated purposes, such as constructing URLs or file name references. You can use amper variables in metadata, which is described in “Dynamic prompting variables” on page 203.

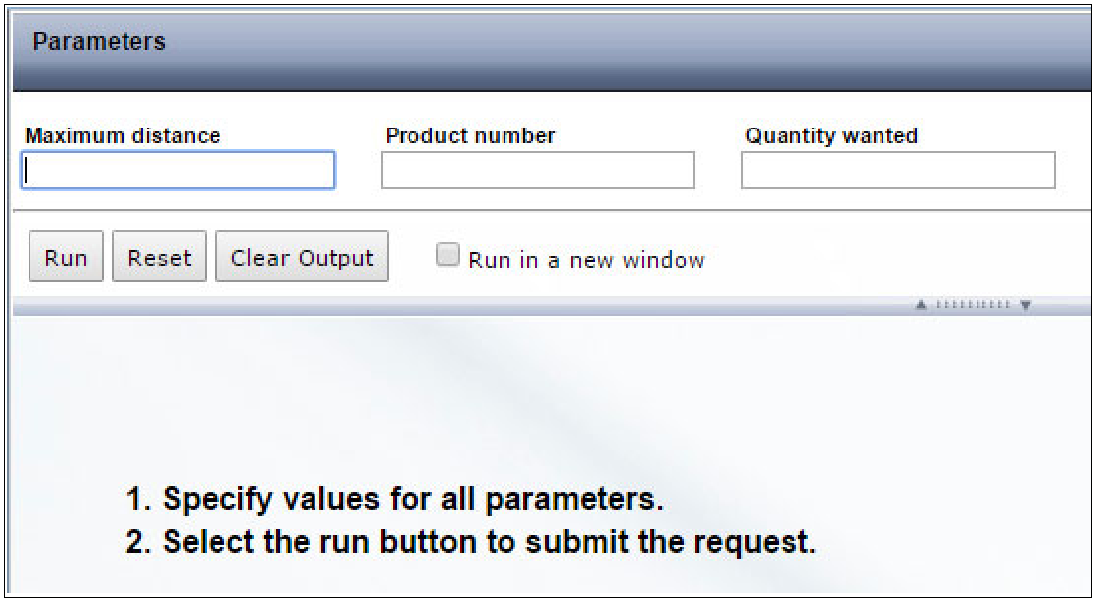
> **Note:** To use the dot file extension separator after an amper variable, terminate the variable with the ‘|’ character (for example, *&DATE|.htm*). Similarly, to use the ampersand character itself, follow the ampersand with the ‘|’ character (for example, *K&|S*).

Dynamic prompting variables

Dynamic prompting variables are local to a report or its metadata. For example, you might want to prompt for a color or model number when reporting on in-stock status of a product. Dynamic prompting variables are defined by the report developer. One or more variables can be defined. The values for the variables must be specified when the report is run. The user who runs the report is automatically prompted for the values, so these variables are also referred to as auto-prompt variables.

When running a report interactively from the BI portal, Developer Workbench, or the WQRAX application extension, the auto-prompt facility in DB2 Web Query identifies the parameters (that is, the amper variables) that are referenced in the report. It then dynamically creates an HTML form to prompt the user to enter the parameter values.

Figure A-3 shows a form that prompts for three parameters. The use case for this example is an inventory application where the user can specify the maximum distance for a store location, the product number of an item, and the quantity needed. The amper variables are declared as Define fields in the metadata. When the report is run, the user is presented with this form.



1. Specify values for all parameters.
2. Select the run button to submit the request.

Figure A-3 Three parameters

It is not feasible to dynamically prompt the users when a report is distributed through a Report Broker schedule. However, the values for auto-prompt variables must still be supplied, which t is accomplished by using the Parameters icon, which is shown on the Schedules ribbon in Figure A-4, to specify the values.

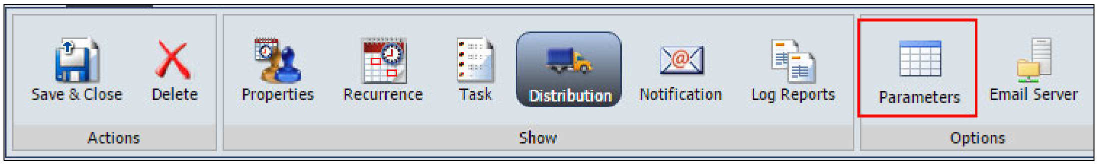
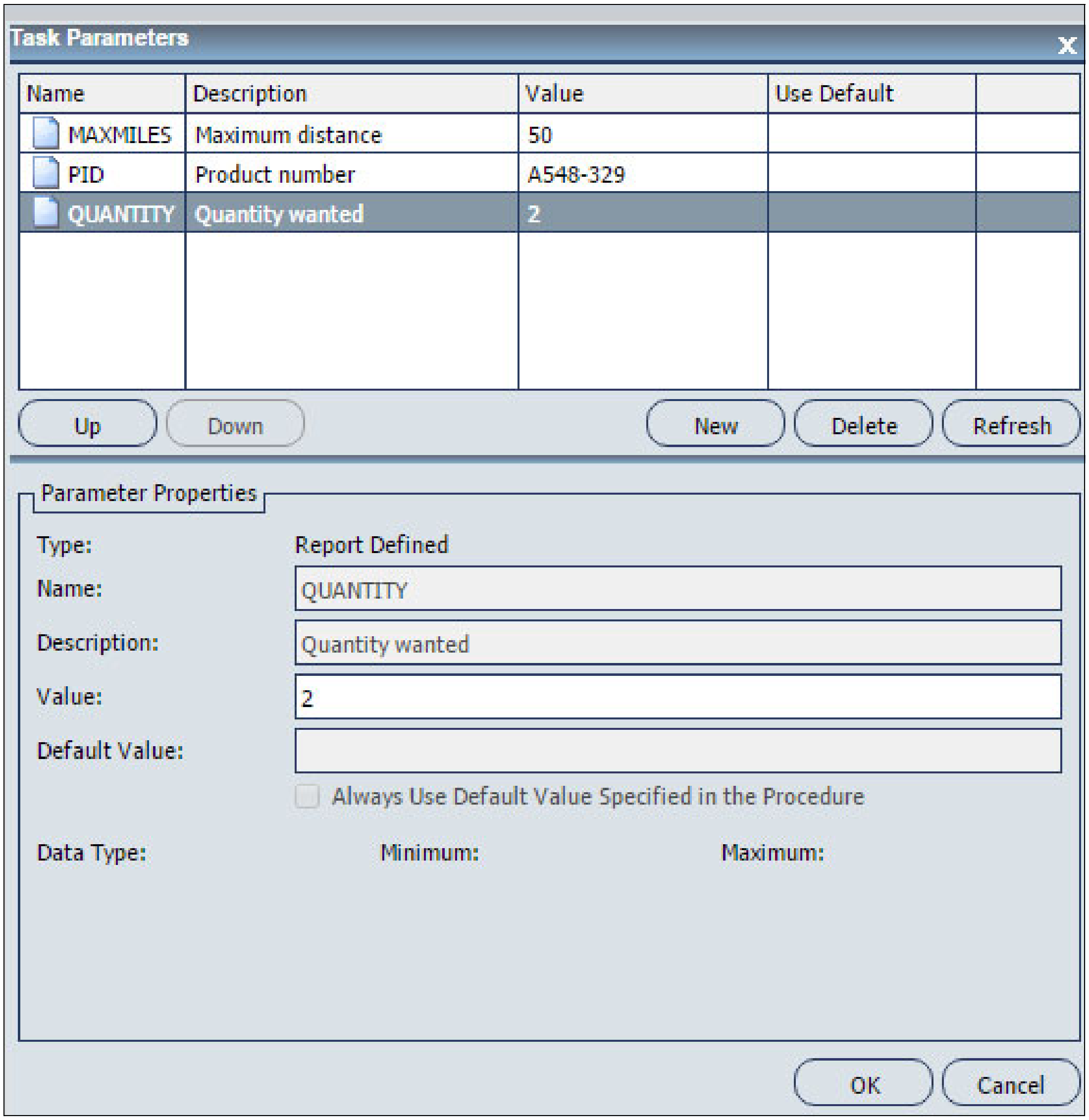


Figure A-4 Parameters icon

When selecting the icon, the Task Parameters form appears and lists the variables that are identified for the report. You can enter a value for each one. The values that you enter are passed to the reporting server each time the schedule runs. Figure A-5 shows a form for the three example variables.



Name	Description	Value	Use Default	
MAXMILES	Maximum distance	50		
PID	Product number	A548-329		
QUANTITY	Quantity wanted	2		

Type:	Report Defined
Name:	QUANTITY
Description:	Quantity wanted
Value:	2
Default Value:	

□ Always Use Default Value Specified in the Procedure

Data Type: Minimum: Maximum:

Figure A-5 Three variables example

If an auto-prompt value is missing when a schedule runs, the schedule fails with a No report to distribute error.

Note: To append a descriptive label to an auto-prompt variable, enclose the text in periods (for example, *&MAXMILES.Maximum distance.*).

Redbooks

IBM DB2 Web Query for i: The Nuts and Bolts

(0.2”spine)
0.17”<->0.473”
90<->249 pages

SG24-8379-00

ISBN 0738442488

Printed in U.S.A.