DB2 Information Management Software

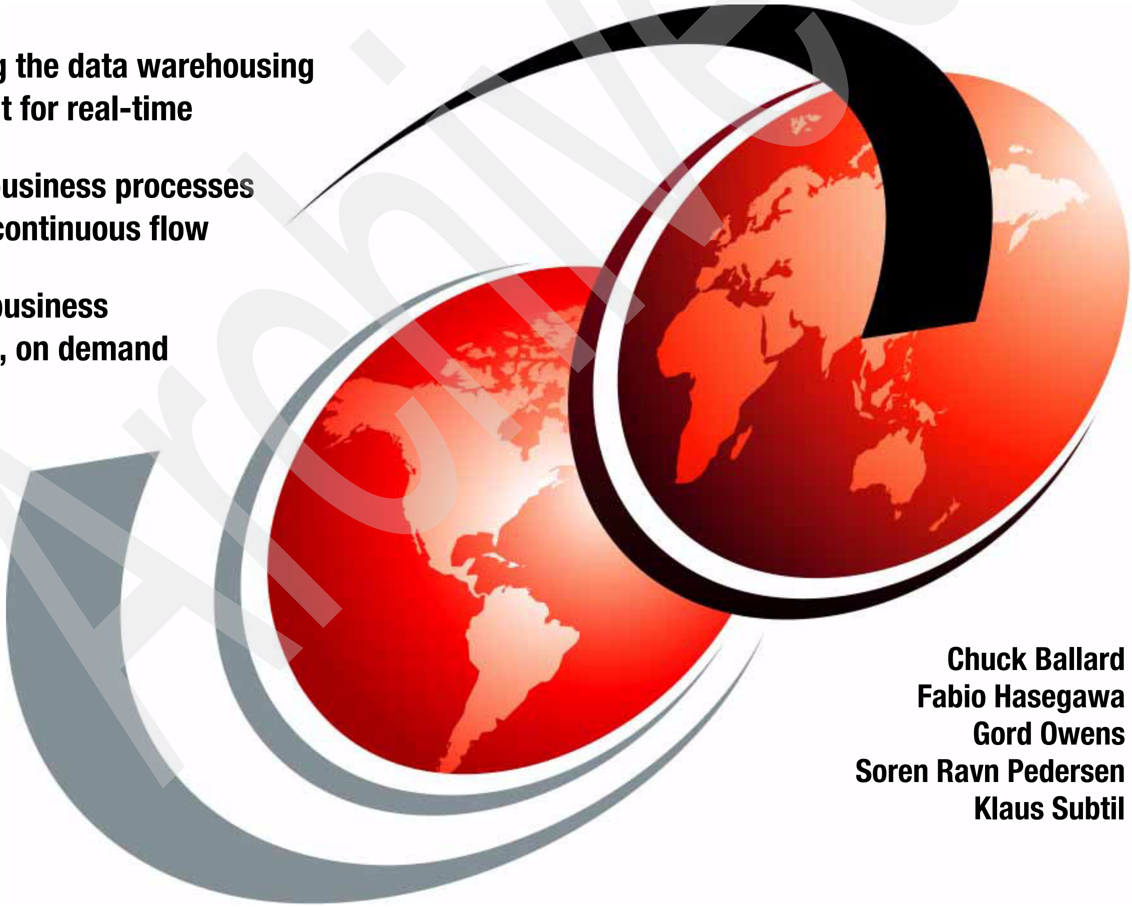
IBM

Moving Forward with the On Demand Real-time Enterprise

Architecting the data warehousing environment for real-time

Modifying business processes to support continuous flow

Delivering business intelligence, on demand



Chuck Ballard
Fabio Hasegawa
Gord Owens
Soren Ravn Pedersen
Klaus Subtil

Redbooks

ibm.com/redbooks

International Technical Support Organization

Moving Forward with the On Demand Real-time Enterprise

September 2006

SG24-7104-00

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

First Edition (September 2006)

This edition applies to DB2 Data Warehouse Edition V8.2 and V9.1, DB2 UDB V8.2, WebSphere Information Integrator V8.2, WebSphere Message Broker V5, WebSphere Message Broker V6, WebSphere MQ V5, WebSphere MQ V6, WebSphere DataStage V8, WebSphere Application Server V5, WebSphere Portal Server V5, DB2 Alphablox V8.2, Oracle 10g, SQL Server 2000, and VSE.

Contents

Notices vii
Trademarks viii

Preface ix
The team that wrote this redbook x
Become a published author xii
Comments welcome xii

Chapter 1. Introduction 1
1.1 The environment today 2
1.2 The on demand real-time enterprise - an overview 6
1.2.1 Information as a service 10
1.2.2 Moving to real-time 23
1.3 Contents abstract 27

Chapter 2. The real-time enterprise 29
2.1 A bit of history 30
2.2 In search of the real-time enterprise 36
2.2.1 Attributes of the real-time enterprise 37
2.3 Positioning business intelligence 43
2.4 Closing the loop 48
2.5 Industry examples 54
2.5.1 Retail 56
2.5.2 Insurance 64
2.5.3 Telecommunications 68
2.5.4 Travel 70

Chapter 3. Architectural considerations 73
3.1 Introduction 74
3.1.1 The impact of data warehousing 75
3.1.2 Information management challenges 77
3.1.3 Creating an information infrastructure 79
3.1.4 The layered data warehouse architecture 81
3.2 Approaches for data access, transformation, and movement 84
3.2.1 Extract, Transform and Load - ETL 84
3.2.2 Enterprise Information Integration 85
3.2.3 Enterprise Application Integration 88
3.2.4 Data replication 90
3.2.5 Custom build 90

3.2.6 Combined approach 91
3.3 Other considerations 93
3.3.1 Database 93
3.3.2 Master data management 95
3.3.3 Analytic applications 99
3.3.4 Closed-loop environments 102
3.4 Information as a service 103
3.4.1 Service Oriented Architecture 103
3.4.2 Enterprise service bus (ESB) 106
3.4.3 ESB and the real-time enterprise 108

Chapter 4. IBM technologies supporting real-time 111
4.1 DB2 Data Warehouse Edition 112
4.1.1 Data Warehouse Edition V8.2 112
4.1.2 Data Warehouse Edition V9.1 117
4.2 DB2 Alphablox 148
4.2.1 DB2 Alphablox applications 151
4.2.2 Deploying DB2 Alphablox 158
4.2.3 DB2 Alphablox services 161
4.2.4 Blox server/client structure 165
4.3 WebSphere Information Integration 167
4.3.1 WebSphere Information Integrator 169
4.3.2 WebSphere DataStage 179
4.3.3 WebSphere ProfileStage 181
4.3.4 WebSphere QualityStage 185
4.4 Process Integration 186
4.4.1 WebSphere Business Modeler 187
4.4.2 WebSphere Business Monitor 195
4.4.3 Architecture 196
4.4.4 Component details 199
4.4.5 Databases 201
4.4.6 The Monitor dashboards 204
4.4.7 WebSphere Process Server and Integration Developer 208
4.5 Application Connectivity 214
4.5.1 WebSphere Message Broker 215
4.5.2 Information distribution 216
4.5.3 WebSphere MQ 217
4.6 Balanced Configuration Unit 217
4.6.1 Balanced Partition Units 219

Chapter 5. More real-time enterprise enablers 223
5.1 Continuous flow processing 224
5.1.1 SQL replication 225

5.1.2 Q-replication 226
5.1.3 Event publishing 227
5.1.4 Comparing the technologies 228
5.2 Messaging 236
5.2.1 WebSphere MQ and continuous flow 236
5.2.2 DB2 MQ Listener 239
5.2.3 DB2 MQ functions 242
5.3 SQL generators and applications 245
5.3.1 Capture 245
5.3.2 Delivery 247
5.3.3 Transform 249
5.3.4 Apply 251
5.3.5 ELT in an enterprise business intelligence system 252
5.4 Extract, transform, and load (ETL) 252
5.4.1 Using ETL tools 253
5.4.2 A WebSphere DataStage project 255

Chapter 6. The project test environment 261
6.1 Preparing for the case study 262
6.1.1 Business functional areas 262
6.1.2 The big picture 265
6.2 Test Scenario A 266
6.2.1 Software overview 266
6.2.2 Test environment 268
6.3 Test Scenario B 271
6.3.1 Software overview 271
6.3.2 Test environment 272
6.4 The real-time enterprise - a case study example 274
6.4.1 The scenario 278

Chapter 7. The case study 281
7.1 Introduction 282
7.1.1 The company 282
7.1.2 The data flow 283
7.2 Tested technologies, results and findings 287
7.2.1 Enterprise Information Integration 288
7.2.2 Extract, transformation, and load 314
7.2.3 The dashboard 317
7.2.4 Enterprise Application Integration 324
7.2.5 Test with combined EII and ETL 340
7.2.6 Closing the loop 350

Glossary 351

Abbreviations and acronyms 357

Related publications 361
IBM Redbooks 361
Online resources 361
How to get IBM Redbooks 362
Help from IBM 362

Index 363

Notices

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

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

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

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

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

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

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

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

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates 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. 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 IBM's application programming interfaces.

Trademarks

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

eServer™
iSeries™
z/OS®
zSeries®
Ascential DataStage®
Ascential®
AIX®
ClearCase®
Cloudscape™
Cube Views™
CICS®
Database 2™
DataStage®
Distributed Relational Database Architecture™
Domino®
DB2 Connect™
DB2 OLAP Server™
DB2 Universal Database™
DB2®
DRDA®
Informix®
Intelligent Miner™
IBM®
IMS™
Lotus Notes®
Lotus®
MQSeries®
Notes®
Rational®
Redbooks™
Redbooks (logo) ™
Tivoli®
Trigo®
UniData®
UniVerse®
WebSphere®
Workplace™
XDE™

The following terms are trademarks of other companies:

EJB, Java, JavaScript, JDBC, JDK, JRE, JSP, JVM, J2EE, Solaris, Streamline, Sun, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Excel, Internet Explorer, Microsoft, Visio, Windows NT, Windows Server, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

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

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

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

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

Preface

We are well into the information age. The advent of the Internet, the growth in the movement to online commerce, the advancements in technology, and the increased online user community has solidified the direction. This has required many changes in business strategies and directions. But businesses have embraced that change, and are feverishly working to position themselves to be successful. It is not really a choice; it is an imperative.

This movement has brought with it the requirement for speed and flexibility. And, there is a need for fast decisions and fast changes, to keep up with, or ahead of, the competition. In these times, information has become one of the most valuable assets required for a business advantage, success, and perhaps even survival. The initiative that supports access to, and use of, this information is business intelligence (BI). And it requires current information to be effective. Thus, the movement is toward a real-time BI environment.

But, this is not a simple move because it requires more than fast access to data. That is, you must first have the data available to access. To have the data to access, in a very low latency environment and at a reasonable cost, requires a robust IT infrastructure. And this, in turn, requires current technology for data storage, application development, and transaction processing. One such technology is based on treating information as a service. This is a very key technology that can impact the entire IT environment, and one that requires a new way of thinking. But the benefits are huge!

Such a new way of thinking will, of course, require an architecture to guide you along the way. That architecture is called a service oriented architecture (SOA), and will be discussed and described throughout this IBM® Redbook. From a business perspective, it also requires building your processes to be event-driven. This will in-turn enable them to be automated more easily and support a more continuous flow processing paradigm. Doing so, moves from batch-oriented processes to real-time, for the processes and the data!

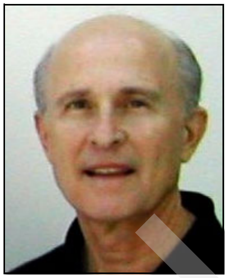
In addition, it requires change to the business, and business processes, themselves to support the IT infrastructure. In short, it means changing the way you run the business. IBM refers to a business that can re-engineer itself in such a way, as a real-time enterprise. Are you in a position to begin moving toward such an environment?

The objective of this redbook is to discuss and demonstrate these current techniques and technologies, along with the strategies, architectures, and capabilities for implementing a real-time enterprise (RTE). To do so, we describe a real-time enterprise, present appropriate architectures, and describe the required components that can support it.

Get ready for the future, because it is now!

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, San Jose Center. The team members are depicted below, along with a short biographical sketch of each:

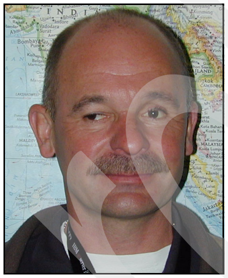


Chuck Ballard is a Project Manager at the International Technical Support Organization, in San Jose, California. He has over 35 years experience, holding positions in the areas of product engineering, sales, marketing, technical support, and management. His expertise is in the areas of database, data management, data warehousing, business intelligence, and process re-engineering. He has written extensively on these subjects, taught classes, and presented at conferences and seminars worldwide. Chuck has both a bachelor's degree and a master's degree in Industrial Engineering from Purdue University.

Fabio Hasegawa is a Senior IT Specialist leading the DBA Distributed Services Center, IBM Application Services, Brazil. He has been mastering several IBM products, such as IBM DB2, WebSphere Application Server, WebSphere Message Broker, and WebSphere Information Integration. He has been working for more than 10 years in IT. During this time, he worked on several projects helping Brazilian clients in various segments (telecommunications, financial, government, and banking). Fabio's areas of specialty are business intelligence solutions, infrastructure sizing, performance management, and design for improving high availability solutions that are focused on information management services.

Gord Owens is a Technical Analyst with IBM Canada. He has five years experience with DB2®, WebSphere®, WebSphere Information Integrator, Query Patroller, and DB2. He has been a Service Consultant for both WebSphere Information Integrator and Query Patroller, and has created and delivered education sessions about product use and problem diagnosis. Gord has studied both Computer Science and Psychology.

Soren Ravn Pedersen is a Certified Consulting IT Specialist, working as a pre-sales specialist with IBM Software Group in Denmark. Soren has 16 years of experience in the IT industry working with development, project management, pre-sales and customer service, primarily in the Information Management and Business Intelligence solution areas. His expertise covers distributed DB2, Informix® and WebSphere Information Integration product portfolios. Other specific areas of expertise and interest include high availability architectures and services enablement technologies in IBM Information Management software products. Prior to his six years of employment with IBM, Soren held various technical sales and management positions with Informix Software.



Klaus Subtil is an IT Specialist within the IBM DB2 Business Partner Technical Enablement Team. He has more than 15 years experience in the IT industry focusing on relational database technology and application development. Klaus is an IBM Certified Solutions Expert for Universal Database 8.1 database administration, and holds a master's degree in Business from the Open University Business School, Milton Keynes, UK. In his current position, Klaus supports independent software vendors and system integrators for IBM information management software. He also certifies students in DB2 at universities that participate in the IBM Scholars Program.

A special thanks to the following people for their contributions to this project; they provided significant input to this redbook project in the form of advice, guidance, demonstrations, and written content:

Rick Cole
Solutions Developer, Business Intelligence Analytics and Solutions, Cupertino, California

Bill O'Connell
Distinguished Engineer, and Business Intelligence Chief Technology Officer, at the Toronto Lab in Toronto, Canada

Paulo Pereira
Chief BI Technologist and Field Advocacy Lead for BI Best Practices, from Dallas, Texas

Jon J. Rubin
Senior Product Manager, DB2 Data Warehouse Edition, at the Silicon Valley Lab in San Jose, California

Paul Wilms
Information Integration Technology Solutions, at the Silicon Valley Lab in San Jose, California

Melissa Montoya
DB2 Information Management Skills Segment Manager, IBM Menlo Park, California

Mary Comianos - Operations and Communications
Deanna Polm - Residency Administration
Emma Jacobs - Graphics
Gabrielle Velez - Editor
International Technical Support Organization, San Jose Center

Become a published author

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

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

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

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

  ibm.com/redbooks

- Send your comments in an email to:

  redbook@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

1

Introduction

In this chapter the primary objective is to give you a high level description of the on demand real-time enterprise, and related terminology. We also give you a brief abstract of the contents of the redbook to help you determine and prioritize your reading selections. Then in Chapter 7, we present a case study that we developed to demonstrate an on demand real-time enterprise in action.

Now, exactly what is an on demand real-time enterprise and why would you want to move your business in that direction? And if you choose to do that, where would you start, and how can IBM help? To get those answers, and more, continue reading.

1.1 The environment today

We find ourselves today in a time where the stock market is very volatile, there is a general slow down in business and we are constantly reminded of the threat of recession, many businesses are struggling to get profitable, and customer loyalty seems a thing of the past. To combat these conditions, companies need capabilities that can enable them to proactively manage and direct their business. One such critical capability required for this enablement is for their information to be closer and closer to real-time, and available when it is demanded. Having information that is more and more current cannot only enable a significant business advantage, but in these times it is fast becoming a requirement for business survival. Further fueling this requirement are the demands for regulatory compliance.

Having real-time information can enable a business to employ proactive decision-making for business goal and measurement achievement, rather than the more common reactive problem impact minimization when it is recognized that goals and measurements will not be achieved. Being proactive is the key. Thus the need for, and focus of this redbook are on the on demand, real-time enterprise.

The topic of an on demand real-time enterprise brings with it a very wide scope, and one that we cannot completely cover in this redbook. So we focus in on the specific area that is of greatest interest to us in the writing of this book, which is the integration with, and support of, business intelligence.

The word real-time is a bit of a non-specific term these days. Not because it has no specific meaning, but because it has been so used - and mis-used. It is now commonly used to mean whatever one wants it to mean - within some reason. Thus is the advent of terms such as near real-time, right time, almost real-time, just-in-time, active, and so on.

Therefore, we give you here a brief description of a few of the terms as they will be used in this redbook. It will help as you read this book, and as you develop your real-time enterprise strategy and direction.

Business intelligence

One characterization of business intelligence (BI) is that it is insight gained from analyzing quantitative business data to make a better informed decision. The typical business intelligence user performs a decision support role and uses business data to provide insight into, as an example, company strategy and direction. In addition to traditional BI, Business Performance Management (BPM) is an initiative that has expanded the scope of BI to include the integration of processes, methodologies, metrics and technologies for the enterprise to better measure, monitor and manage business performance.

By contributing to an integrated, enterprise wide data and reporting approach, and by providing highlights into the business direction, BI is able to provide management with the information they need and at the time they need it. Technologies such as Web services (to minimize the time and cost of application development), federation and information integration (for heterogeneous data accessibility and data integration), grid computing (to supply the resources needed, and enable the support of large volumes of data), and interactive portals and dashboards can enable the use of a broader range and variety of more current data, presented on demand and in a format that is much easier to interpret and understand by a larger number of users.

The primary focus of BI is to develop a cross-process reporting strategy through data analysis and decision support capabilities. One step in this direction is to develop common dashboard tools and a common dashboard infrastructure. Through dashboards, analysts can reduce the time spent collecting data and increase the time spent analyzing it.

As BPM becomes more integrated with BI, it is important that BI expands beyond business decision making to also provide closed-loop feedback to enable improvements in the business processes.

The BI environment is changing in other ways as well. For example, companies no longer have the long strategic time-frames in which to plan, design, and manage processes. Business managers are looking for answers to their questions much more quickly than in the past. Strategic time-frames have continued to become smaller. For example, yearly revenue goals and measurements have, for most enterprises, become quarterly goals and measurements. Investors and share-holders are more demanding and more critical of missed performance goals, even with these shorter measurement periods. And, they are expressing their opinion by giving, or withholding, their investment money.

These demands are coming at a time when the volume of data is growing, there is an increase in business mergers and acquisitions, the use of strategic outsourcing is growing, and there is an increasing requirement for faster and faster turnaround on information requests. This has put an enormous burden on the information technology (IT) organizations. And most of this change is centered around BI, because that is the environment responsible for providing information for decision-making.

Business Intelligence is the process by which you can obtain accurate and consistent business data from your data warehousing environment, analyze these data from various business contexts, identify trends, variations, and anomalies, execute simulations, and obtain detailed insight (intelligence) about your business. Having this intelligence enables faster and easier identification and resolution of business problems.

Having right-time business intelligence offers the opportunity for proactive management of the business. For example, you can identify and be alerted to potential problems that could hinder meeting your business goals and measurements. And you have the required information for decision-making, and for taking appropriate actions to avoid them. This result can be improved business performance management, and improved business process management.

Real-time

We think that the term real-time can be used with greater meaning, and in a higher context, when used to describe an enterprise. So we will focus on describing the term real-time as it applies to the real-time enterprise.

First we look only at the term real-time, and specifically how it is used with reference to the information flow in business processes. In general, real-time means that you can know the result of an action or event in a process, instantaneously after it happens. But instantaneously is very fast and very difficult to achieve in most situations. So there are many caveats in this statement. Therefore we want to provide a bit more freedom and flexibility when using the term real-time, particularly as we discuss it in a specific context such as the real-time enterprise.

Right-time

Because there are caveats with the term real-time, we have chosen to introduce a new term. And, to do it in the context of BI. It is clear that not all information needs to be delivered in real-time. And even if there is a perceived need, cost justification must be factored into the decision. In most situations, the information is needed, when it is justifiably needed. And, the currency level of that data should be such that it enables appropriate decision-making. The term we have selected to indicate this requirement is *right-time*. That means, the right data is delivered to the right person, at the right time, and at an acceptable cost to enable the required decision-making. This is right-time BI.

> **Attention:** As a point of reference, we say that **right-time BI** means that the decision-maker will have the most current and accurate information that is necessary and cost justifiable, at that time, to make an informed and intelligent business decision.

Real-time and the enterprise

We stated, in the Preface of this redbook, that we are well into the information age. As such, there is now an even greater need for more and current information to enable fast decisions and fast changes, to keep up with, or ahead of, the competition. And now we have called this capability right-time business intelligence.

Although it is closely related, we now want to describe the real-time enterprise. Obviously in a real-time enterprise, we want to have information in, or close to, real-time and be able to support right-time business intelligence. We think if you can do this, you have a real-time enterprise.

But, it is not quite that simple. So we have more caveats. For example must a real-time enterprise have this, or just be committed to, and moving forward with, it? And, must it exist in all business functional areas? The short answer, is no.

Attention: In simple terms, and for purposes of this redbook, we say that an enterprise that is committed to, and moving forward with, real-time processing and business intelligence, in those business areas where it is required and financially justified, is a **real-time enterprise**.

Moving forward

IBM has long been a leader in strategic product planning and development, and provides robust frameworks and architectures for supporting initiatives. They have invested time and money in research and product development to enable the flexibility and capability to grow and change with requirements and technology. This ability to look beyond current requirements in their planning is a prime differentiator from the competition. And IBM has applied that same approach to constructing the architecture for the real-time enterprise.

We understand that the descriptions presented here are a bit loose, but they tend to be more practical, and fit the requirement for use in this redbook. And, hopefully, they are more easily understandable and usable. And, most of all, they should better enable you to meet your business measurements and goals and give you a business advantage.

There is related information about this topic in another redbook, “Preparing for DB2 Near-Realtime Business Intelligence, SG24-6071. To get this, and other, redbooks, see “How to get IBM Redbooks” on page 362. The need for real-time, or right-time, information now permeates the entire enterprise. Thus the need for an expanded solution that addresses the requirements of the enterprise.

Are you ready for more about the real-time enterprise, and right-time BI?

1.2 The on demand real-time enterprise - an overview

An on demand business needs to extend its information management strategy by transforming its business processes and changing how the company uses information to drive insight. For example, the integration of heterogeneous data sources is fast becoming a basic requirement to satisfy the needs of the business. This will enable you to adapt to the changing environment, and leverage your data for a business advantage. It will also provide employees and applications with easy access to an integrated source of enterprise information.

But do we really need a real-time enterprise? Do we need access to all the data in the enterprise instantaneously, or at least very quickly?

Unfortunately the answers are either yes, no, maybe, and sometimes. And then the ever popular follow-up of *it all depends*. And one of the things upon which it depends is your definition of a real-time enterprise.

Instantaneous access or current data?

But, hold on just a minute. We have already arrived at another point of contention. Real-time does not simply mean having access to the data instantaneously after some event. Real-time data is current data relative to a particular business event. So real-time is relative to the completion of the event, rather than the speed of access to it.

For example, say an event only happens once a day. The data about that event can still be considered real-time anytime within a 24-hour period, because it is still the most current data. However, we agree that the ability to have that data as quickly as possible after the event will make it even more valuable. So, we can agree that this is about semantics. The message is, do not be overly concerned about the semantics. Be more concerned about the functionality, practicality, and cost justification of the environment.

More than data access

You should understand that to become a real-time enterprise requires more than simply having instant access to current data. You must implement support to make sure current data is really current!

Basically, a real-time enterprise is similar to what we at IBM have referred to as an on demand enterprise. Here is another point of reference.

> **Attention:** An **on demand enterprise** can be thought of as one whose business processes—integrated end-to-end across the enterprise and with key partners, suppliers and customers—can respond with the appropriate speed to any customer demand, market opportunity, or external threat.

However, that description implies integration and action. Integration is all about enabling people and applications to interact with other people and applications. That lets you leverage information, based on end-to-end business processes within and beyond the enterprise. It requires business flexibility, agility and IT simplification. Therefore, using the terms in combination is very appropriate. That is, we now use the term *on demand real-time enterprise*. A high-level view of such an on demand real-time enterprise is depicted in Figure 1-1.

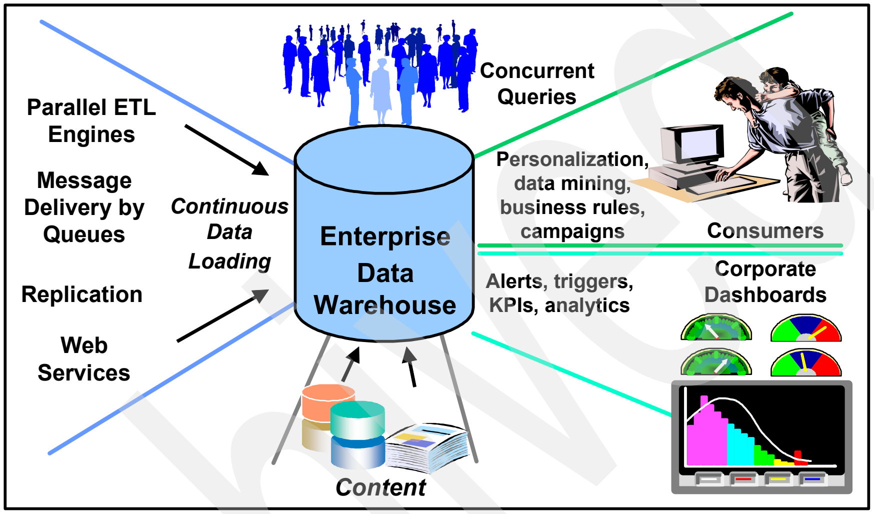


Figure 1-1 On Demand Real-time Enterprise - an example

The enterprise data warehouse (EDW) can be continuously updated by processes such as extract, transform, and load (ETL), extract, load, and transform (ELT), messages that are passed via queues, replicated data from tables or queues, and Web services. The data can be both structured data and un-structured content. However, this activity does not impact the continuous access to the data warehouse by such processes as concurrent queries, analytic applications, data mining, and corporate dashboards.

When we talk about the EDW, we are typically talking about an enterprise data warehousing *environment*. That is because the environment consists of the EDW, operational data store (ODS), as well as physical and virtual data marts. Including the ODS in the data warehousing environment enables access to more current data more quickly, particularly if it happens that the data warehouse is updated by one or more batch processes rather than continuously.

The following list contains a few more things you will need to consider as you move towards an on demand real-time enterprise:

- **Standardize business processes:** In addition, they need to be integrated and managed so that they enable more of a continuos flow from process to process. They also need to support online monitoring, and be capable of having their status reported upon completion - by either a manual or an automated activity. In addition, they must be flexible and dynamic in nature to enable them to be changed quickly and easily as the business changes.
- **Enable information as a service:** This implies a number of actions. As examples, moving to the use of Web services and developing a service oriented architecture (SOA). Here, business functionality is developed as a number of application modules, or services. These services can be shared, used, and reused, making application development faster, simpler, and more cost effective through code standardization and reuse.
- **Information integration:** By enabling applications to access data from heterogeneous sources and operating environments, whether local or remote, you can begin to standardize data access and promote consistency across the enterprise. Web services promote standardized data access through the use of a base of standard services by the enterprise.
- **Connect participants to business processes . . . in real-time:** Participants in the process need to have access to enable real-time (instantaneous) monitoring and reporting on the status of the business processes. Participants can be humans or applications. And from an applications perspective, you should be moving towards more use of in-line analytics. That is, increased use of automated analytics that are embedded in the applications, rather than requiring human intervention, to minimize or eliminate delays in decision-making time.
- **Provide real-time analysis and insight:** Having real-time, instantaneous, data is of little use unless it can be accessed and analyzed. That means the data has to be made available to users and applications from wherever you have it stored, to enable analysis and right-time business intelligence.

For more detail on these considerations, see 1.2.2, “Moving to real-time” on page 23.

So you have some choices, and you will have to decide what your particular requirements are for becoming a real-time enterprise. In general, however, you should understand that moving in the direction of a real-time enterprise can bring you significant business advantage. Most companies understand the advantage, and agree that those moving the quickest will reap the most rewards.

Integrated information infrastructure

To support the real-time enterprise, you will need to evolve towards an integrated information infrastructure. This is discussed and described throughout the redbook, but we give a high level view here to get started on the journey.

An information strategy must be built on an extensible foundation with rich capabilities supporting innovative business applications. Such a strategy implies an information infrastructure. That can simplistically be thought of as having two parts, the information assets and the information services necessary to support innovative business applications. The information assets can include information managed in databases, file systems, data warehouses and content repositories. Robust asset management can include critical capabilities such as sharing, performance, scale, availability, and security.

An integrated infrastructure implies there should be a framework in place that enables the enterprise data to be integrated, accessed, and used by the enterprise. This not only includes the enterprise data warehouse, but other internal, and even external, data sources that are used. That data can then be used by queries, analytic applications, and operational transaction systems to support the enterprise. This is depicted in Figure 1-2.

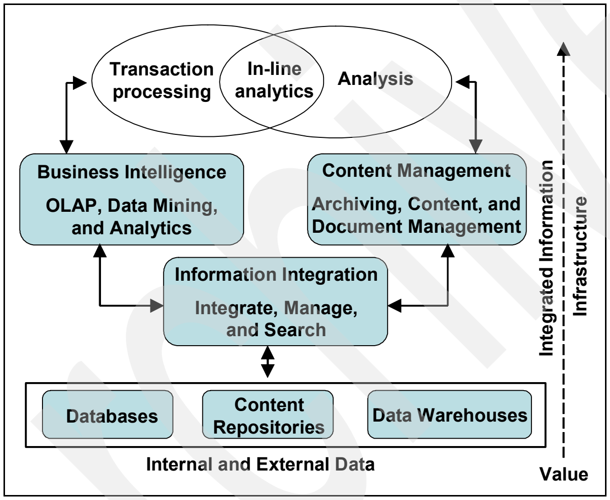


Figure 1-2 On demand business processes

There is a relatively new term used in Figure 1-2, and that is *in-line analytics*. We most typically think of analytics as analyses being performed to discover the root-cause of a problem or issue. And most typically this is performed external to a process. That is, we recognize a problem and then go in search of an answer. We may use search engines, or execute queries on some data environment.

However, the focus now is on a more automated or intelligent approach. Or, at least one that is part of a process. Most likely, it is the same process wherein the problem was recognized. The analysis is then *in-line* to, or part of, the process. And likely it is automated analytics. However, it could also be a *human activity* embedded inside a process. So, this term refers to the integration of analytics with the business processes. It is a key technology in a real-time environment.

1.2.1 Information as a service

A clear direction in the marketplace it towards providing information as a service. Why so much emphasis on information? Because information is power. It is the power to make decisions, the power to gain insight into businesses, the power to provide better customer service, the power to find new customers, and the power to find new revenue opportunities.

IBM has set a clear direction to what is called *Information on demand.* This is all about *getting the **right** information to the **right** people at the **right** time.* It is not a product, it is a strategy, or direction. It enables the creation of business value by integrating, analyzing and optimizing all sources of information throughout the information life cycle, to manage risk, and help create new business insight. A major objective is to reduce the cost and complexity of information availability.

Although information on demand is not a product, it will realized through the implementation of products. And they will provide the information as a service.

What does that mean?

So what do we mean by *information as a service?* In Figure 1-3 you see typical repositories as they appear, either running on DB2 or as a content management application running on IBM Content Manager. There are also disparate data sources from external suppliers and business partners.

As new tools and applications are built, the use of information is needed. And these new classes of applications need not rely on the source of the information. In fact, information sources change every day and the consumption of information changes every day. Therefore, a flexible architecture needs to deliver information as a service, when it is needed and in a standardized way, to different users of the information in the environment.

Information as a service must be flexible - and based on standards, such as XML, (xQuery) and JSR170, and Web services. Information could be delivered in real-time, shown on the right side of Figure 1-3, like an online hub, for example, or as synchronized master data, or as extracted by off-line analysis such as Basel II. So information as a service, delivered more flexibly as the need arises, empowers more users of information in the enterprise. The middle portion of Figure 1-3 references IBM investments over the last few years both in acquisition and organic development.

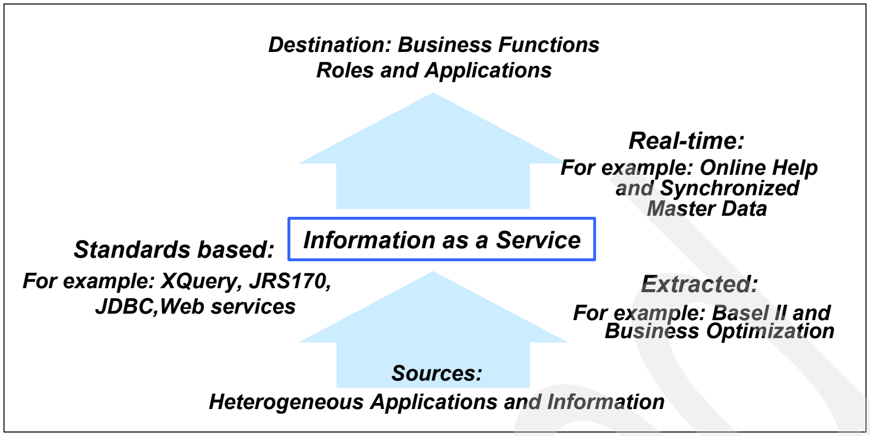


Figure 1-3 Information as a service: Separating information from business logic

Now take a look at how master data is being delivered as a service to multiple applications. Figure 1-4 illustrates the situation. Typically in a large organization you will see different product names and product numbers across the enterprise. And if we want to move, for example, an over-stock of coffee beans in Mexico to an under-stock situation in Texas, you cannot do it right now because the product codes and information do not match

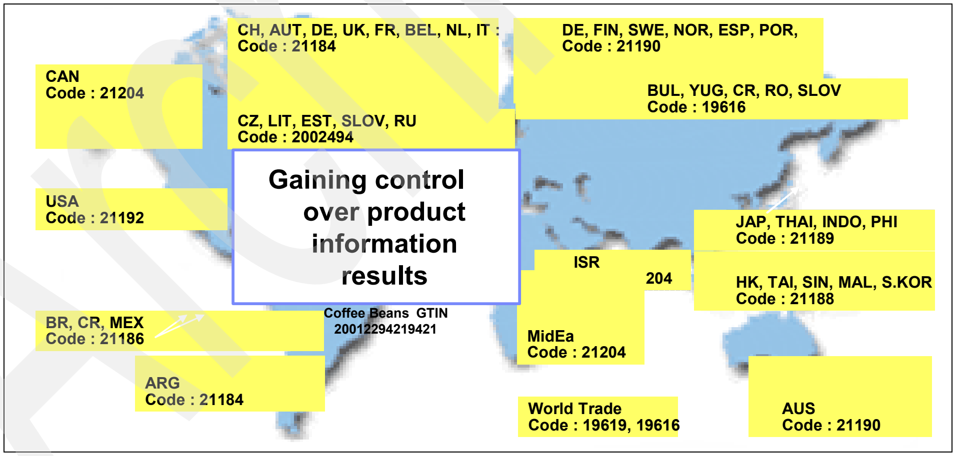


Figure 1-4 Master Data Management: Inconsistent master information

However, by implementing information as a service and matching the appropriate information and delivering the master data, studies have shown that you can get significant improvement in optimized promotion, moving logistics, brand management, reduction in the number of calls to a call center, and many other capabilities. In the example shown in Figure 1-5, the following resulted:

- 27% improvement in optimized promotions
- 23% improvement in maximizing product and brand management
- 27% reduction in call center questions about basic item information
- 20% improvement in employee productivity

The industry drivers were RFID, waste electrical and electronic equipment recycling, product information exchange standards, return of hazardous substances, global data synchronization, Sarbanes Oxley, and so forth.

Master data is not only about item information, but also about understanding customers, understanding a location, understanding an event, understanding the business context of how the data will be used, and not just the integration of data and content.

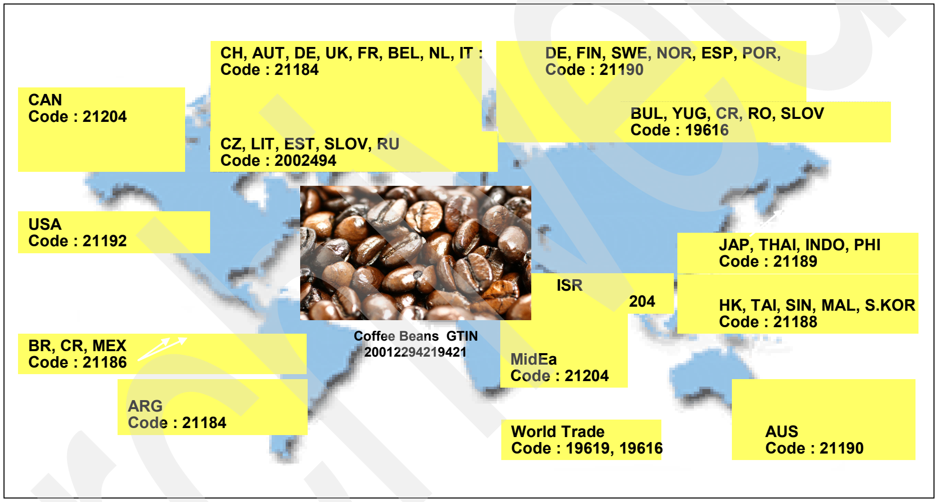


Figure 1-5 Master Data Management: Information as a service

Another example

Now we discuss an example of how one bank merged with another bank and wanted to deliver value in the customer-related content, and maximize the benefit of the merger. They wanted an integrated view of customer information across multiple business units within the bank. Most of this information was unstructured. Customer information was stored in IBM Image Plus, FileNet image repositories and many other environments. We could have moved all the data into a single repository and embarked on two or three projects to do so. But by leveraging technologies and federated information integration services, we were able to provide unified access to distributed content across retail, commercial, and brokerage organizations, and do it very quickly. The result has been improved customer service much more quickly at the retail branches and improved acquisition readiness. Leaving content in place was one-fourth of the cost of consolidation - meaning that if they had consolidated content rather than delivering information as a service, it would have cost them four times more money. This is what be mean by enabling business flexibility with information.

There is a class of information, in terms of application domain, that is emerging. Here are some of the characteristics on which we can focus:

- Manage risk and streamline compliance
- Gain insight through business intelligence
- Drive business transformation
- Gain control over Master Data
- Manage information complexity
- Lower cost through optimized infrastructure

You already know that we can gain insight through business intelligence, and the associated analytics used to query and analyze the data in the data warehouse. But we also provide the core capabilities to consolidate and federate information that can be delivered by means of a service.

This is an area where IBM has made investments and where we are expanding our market. As shown in Figure 1-6, existing applications can be information enabled. So when information is available as a service, the applications can take advantage of it. For example, consider procurement.

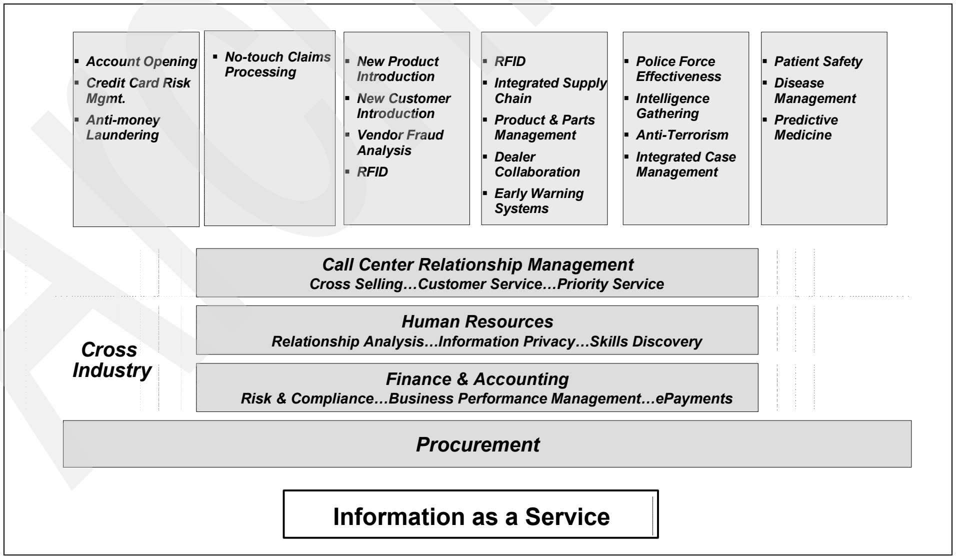


Figure 1-6 Information enabling existing applications

You can improve procurement processes because you know what previously has been done, and have a good understanding of the best practices associated with those processes. You can improve finance and accounting, and human resources management, as well as call center relationship management. These are four key horizontals that are a focus for enabling existing ERP and non-ERP applications to function as an information service.

There are also industry-specific applications, such as finance for example, that may need to be introduced across many different countries, and through multiple stores to many different suppliers. And you can do a much better job if you have the master data integrated with the end ERP application. We are not saying to simply focus on the segments previously mentioned for information as a service. But also consider each of the horizontal and vertical applications, such as those depicted in Figure 1-6, to determine how enabling these existing applications to information as a service can improve efficiency.

Figure 1-7 shows the results of a survey conducted in a number of countries to identify the key capabilities that companies intend to develop to address the business challenges identified. What you see in this figure is what the line of business managers identified, rated as challenges, and then indicated how they intend to develop capabilities to address them. Their solutions center around organizational growth, responsiveness, and effectiveness related to people and processes.

Business Challenges *Top Tiers by Importance*	**Capabilities** *Top Tier by External Spending*	**Business Impact Area** *Most Selected*	*2nd Most Selected*
▪ Streamline/improve efficiency of business processes	▪ Introduce new apps and systems into existing IT infrastructure ▪ Standardize, automate and integrate business processes, apps ▪ Align business models, processes and apps to business goals	▪ Information Availability	▪ Operational Stability ▪ Productivity
▪ Better understand and meet customer expectations	▪ Analyze and use information to make better business decisions	▪ Information Availability	▪ Productivity
▪ Increase employee productivity	▪ Enhance employees' skills and expertise continuously ▪ Empower employees with tools and info for decision-making	▪ Information Availability ▪ Operational Stability ▪ Productivity	
▪ Maintain continuity of business operation ▪ Secure business operations, IT infrastructure and privacy ▪ Ensure regulatory compliance and business controls ▪ Improve utilization and management of IT resources	▪ Reduce/mitigate risk associated with business operations ▪ Ensure apps and systems run without service level degradation ▪ Improve utilization of hardware, software ▪ Improve flexibility to increase/decrease IT capacity (apps, systems)	▪ Risk/Reduction ▪ Operational Stability ▪ Productivity	

Figure 1-7 Capabilities survey

By drilling down and soliciting feedback from the line of business executives, the base problem increasingly centered around the concern that employees could not get the information required to do their jobs. This is reflected in Figure 1-7, in somewhat of a tangible form.

What is being reflected in that survey is that the line of business executives talking very specifically about the business challenges they face. Challenges such as streamlining and improving efficiency of business processes, better meeting customer expectations, and increasing employee productivity.

As you can see highlighted in Figure 1-7, the top-rated area by far, representing their biggest business challenges, was *information availability*. The concern around process, the pressure on the line of business to reduce their core process times, the fact that there is information they need to process during those same cycle times, and that the *variety* of information, are all greatly increasing. They also face increasing risk around this information. That is, they are forced to store more information for longer periods. This is for compliance or legal or operational risk reasons, where they manage the information in a very rigorous manner for long periods of time. This, not surprisingly, is resulting in the generation of more versions of the same information, with more versions of metadata, and resulting in more inconsistency and reduced information accuracy.

However, there is the ability to get to a single version of the truth in terms of customers or products. For example, a recent incident with a particular brand of automobile tires, resulted in the Tread Act. This act requires the ability to track a single tire throughout its bill of material and assembly onto the automobile, and finally to the purchaser.

This type of information tracking is becoming increasingly important. This is also true of government service, in terms of the ability to unify all of the government information from a local, state and federal level, including departments such as the Social Security Administration.

An increasing number of large enterprises are starting to launch global events and global activities centered around information management. This is no longer simply a tactical activity, but also a strategic one. And, not surprisingly, organizations have been busy trying to stem this tide, as shown in Figure 1-8. To do this, they are moving to an extensible infrastructure. That is one that:

- Is process oriented
- Is adaptable and enables rapid implementations
- Includes an abstraction layer to protect integration
- Is cost effective
- Uses existing processes, information, and skills

To get to such an environment requires:

- Tapping into existing or earlier systems. These are proven systems with significant resource investments.
- Deploying point solutions. These typically become information silos, resulting in inefficient utilization of resources.
- Consolidating data sources. This is an effort to reduce cost. It is important to do, but is time consuming and brings with it a number of issues of integration. As examples, tactical data silos can still exist, data element definitions can differ, and issues with heterogeneous systems can take significant time and effort.
- Dealing with compliance. Compliance must be implemented, and issues addressed to mitigate the risks involved.

This is a massive market that is being driven by business requirements. But, the move to this extensible infrastructure promises added capability and lower cost. As examples:

- It is a market with maturing technologies
- There are new and emerging standards
- There is a migration to a services oriented architecture

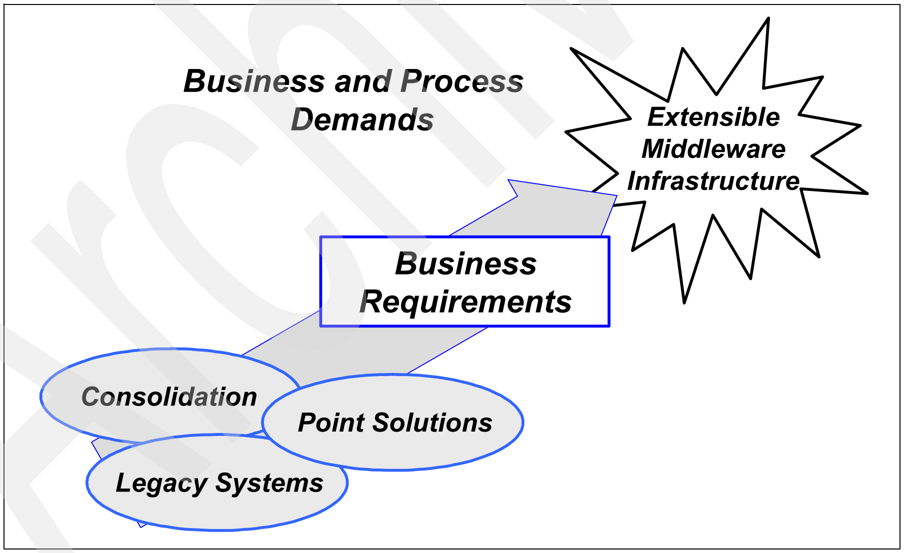


Figure 1-8 Complexity

Will we ever get to the point where everything is consolidated? Perhaps. If you get back to those business and process demands that are facing businesses and the line of business managers, this is what is driving an increasingly accelerating demand.

Technologies have been developed that hopefully deal with some of this from a problem perspective. And we have reached a point now in terms of this technology where it has matured and we are reaching somewhat of a tipping point in the sense that the data and the content which has traditionally been treated very separately from one another is now at a point where you can run a single query or a single search across both sets of information.

That ability to unify full text indexing with text mining has become a reality, and we are now starting to see the benefits. We are also seeing new standards, such as JSR170 and (xQuery) and (SOP) and (Wisdal), that are making it easier to unify both the information and the services. One of the most important initiatives is represented in the center of Figure 1-8, and that is the *service oriented architecture*.

Roughly 20 percent of organizations today are actively engaged in service oriented architecture designs and rollouts. Research indicates, that 20 percent threshold will reach 80 percent by the end of 2006. What is significant is that organizations are stepping back and looking at the complexity that they are facing from an IT infrastructure. The result is a determination to build their next generation of applications in the form of composite-based applications.

They may be deploying these on a problem-by-problem basis over the course of the next decade, but they are developing blueprints and determining which partners are going to help them roll out this next generation of infrastructure and applications. And clearly *information availability* is becoming an important element of this. The right side of Figure 1-8 addresses the benefits of a service oriented architecture. *Information as a service* is a critical part of what organizations do around a service-oriented architecture and information enabling that next generation of applications

So what does it take to do this? Ultimately what we are talking about in terms of information on demand is creating business value by integrating, analyzing and optimizing heterogeneous types and sources of information throughout their life cycle to manage risk and create new business insight. We are talking about enabling with information on demand. If you think about it, ultimately what we are enabling is action, as illustrated in Figure 1-9.

You need to increase business value and business flexibility and responsiveness in an open manner by enabling action -- timely and proper action -- which allows organizations to take advantage of the *windows of opportunity* that they are missing today. This drives competitive differentiation and responsiveness. In order to do this you need to enable the organizations to connect to the various data that they have in their applications, in their data warehouses, in the various content repositories and the like. But it is not adequate just to connect to the data, you have to enable them to integrate that data.

You also have to profile it, cleanse it, reconcile it, and wrap consistent metadata around it. In addition, you have to take a look at the other supporting information, such as e-mails or contracts that reside in a document format or in invoices as an image or maybe a rich media file or report. And, look at all of the information in a holistic manner. That ability, to deliver the information in context to a user or a business process, becomes critical. It is not about just connecting things so you can see 3,000 or 30,000 items. It is about getting four or five of the most relevant pieces of information, regardless of their format or source, to the user, knowing their security and their role in the organization and what they are trying to perform. Better yet is the ability to look at all of this information as a holistic body and look across that to ascertain predictions on what is likely to happen next from an operational perspective.

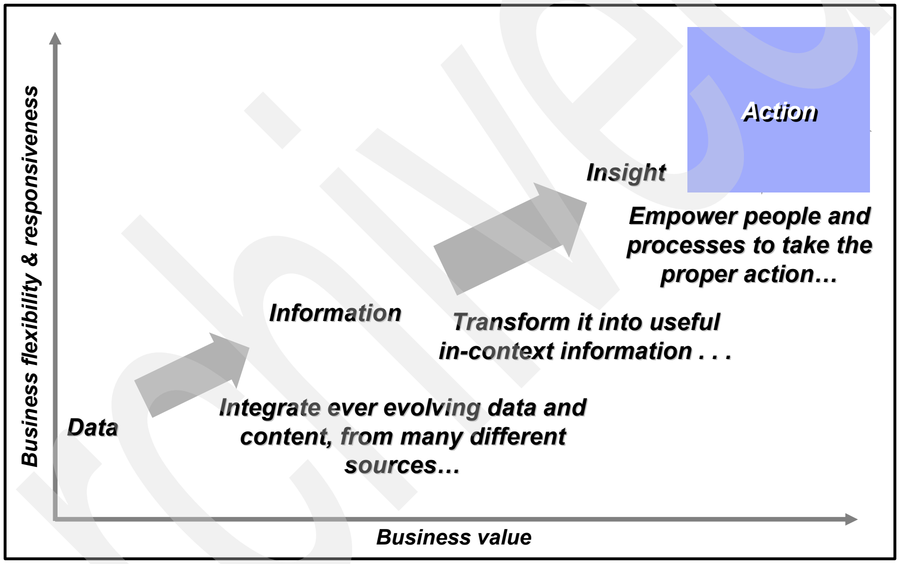


Figure 1-9 Increasing the value of information

That provides insight upon which a user or business process can add even more value. Then you can model those predictions and come up with a series of recommendations based on the prediction. For example, *what is the action that should be taken based on what is likely to happen next*, or *what is getting known as a class of solutions around prescriptive analytics*. This all arms the user with further insight to really drive timely action that in turn drives competitive differentiation.

Technologically, what we need to do to enable this is to embrace and extend what organizations have already invested in. And in some situations that means increasing the value of, say, some heterogeneous database or content repository. It is also about enabling increasing flexibility and choice, such as the ability to look at a merged company and quickly embrace their information sources and make it useful to the organization.

We are talking about increasing the quality and relevancy of all information, improving the integration and access to the information, and harkening back to that vision of information as a service. But we have to do this in a way that it increases organizational responsiveness, productivity and ROI. There has to be a recognized business value. So we have been evolving what you see represented in Figure 1-10, which is the ecosystem that we call *information on demand.*

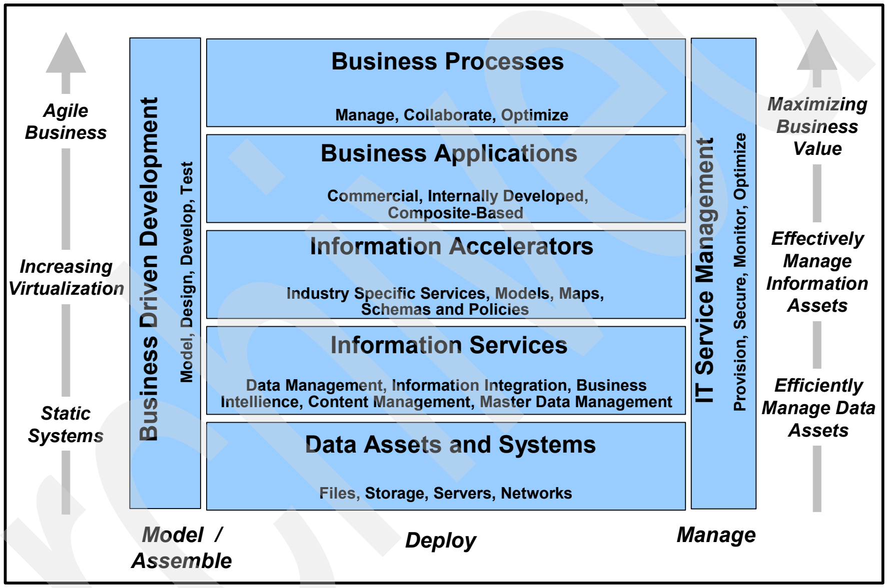


Figure 1-10 Ecosystem of services

The model is all about information enabling business applications and business processes, and it is being done in an open manner.

Notice that the model looks somewhat hierarchical in nature. This is because users tell us they cannot get their arms around it in terms of what it takes to strategically information-enable their applications and business processes. The fact is that some of the services you see represented here are recursive in nature, meaning things move up and down the stack.

You will also notice that we are discussing this in the vernacular of a service oriented architecture because organizations are looking at this strategically in how they are going to build that next generation of infrastructure and composite-based applications - which are services oriented. Much of what you see represented here can be called directly using an API. It is just that strategically we have moved beyond talking about integration as an API level call, to integration as a service.

On the left side of Figure 1-10 you can see the technologies necessary to build, or in the new service oriented model, model and assemble applications and business processes.

At IBM this is the Eclipse-based technology and the Rational® components. On the right side of Figure 1-10 is the IT service management. These are the managed capabilities for provisioning, securing, monitoring and optimizing the business applications, business processes and the underlying infrastructure. This is, as examples, our Tivoli® technology and our tooling technology for optimizing products such as DB2 and Informix Dynamic Server.

In the middle of the figure, you see the deploy layers. The top layer represents the end-to-end business processes. We say end-to-end because this is really what the customer is using to do things such as originate a loan or adjudicate a claim. And the steps they go through today are not technology-enabled.

If you look at the end-to-end process, there are some technology areas where we can add value by replacing manual activities, converting them to digital processes that go into the business process itself. This enables an exception, for example, to be handled by not just using the phone but by using collaboration technologies coming from our Workplace™, or Lotus®, business. Beneath that we have the applications themselves and the techniques to surface information and make it available to the user.

Beneath that we have the raw data assets and systems that support this information. Here we are talking about where the information physically resides. As examples, in the file servers, the storage area networks, the applications servers and the storage networks. And then we have the base which builds out information on demand and differentiates IBM information services and information accelerators.

Getting to information as a service involves virtualization, which is the ability to abstract the lower level services as shown in Figure 1-11. Research shows that somewhere between 30 to 50 percent of the IT budgets are going to integration costs, and the higher level costs are really at the business application and business process layers. And those tend to be fairly dynamic in that any time you want to move to another technology, such as moving from Oracle to IBM, or moving a content repository from FileNet to IBM, or perhaps moving some of the tooling from BMC to IBM Tivoli, it impacts that higher level code.

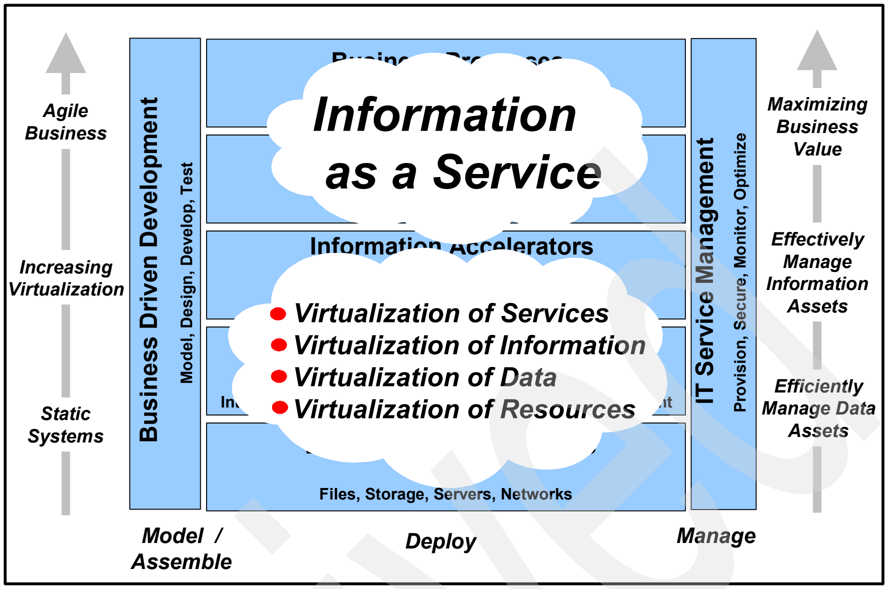


Figure 1-11 Virtualization

We are talking about enabling virtualization of these lower-level resources. As examples, being able to virtualize the physical resources, or the data, or the actual services that call the data. Ultimately, and in combination with standards, this gets you to what you see in Figure 1-11. That is the ability to treat information as a service, not caring where the it came from or what format it is in, but making it available to a user or a business process or an application as it is needed.

The standards have to evolve here and IBM is in the forefront helping in that evolution. This is a critical concept in terms of making the technology available in an open manner, so that our customers can embrace it.

The heart of what we are discussing is what you see in Figure 1-12. This is really what sets IBM apart. It is a set of services to information enable applications and business processes to accelerate the actual solutions to solve those business problems.

At the lowest layer in Figure 1-12, you see the data management services. This includes technology from IBM such as U2 pervasive devices, the open source Cloudscape™ and Derby, and Informix.

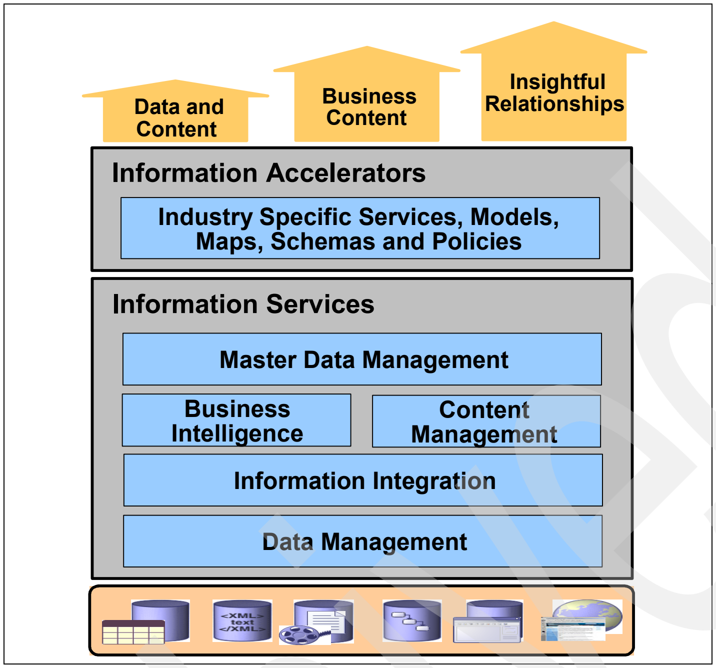


Figure 1-12 Information services and accelerators

It is important to note that the data management world is about to evolve from second generation relational databases to third generation hybrid relational databases. It will fundamentally change the landscape of data management as it exists today.

There is a huge evolution at this layer, and information integration is at the core. This is the primary element in terms of connecting, enhancing (that is, profiling, cleansing, reconciling and wrapping that consistent set of metadata around it), and the ability to search across all forms of information and publish it to business processes, applications, and users.

It is also about the business intelligence layers, where data access, analysis, and reporting take place. And it is about IBM content management, which is now completely Web service enabled.

This is where the IBM acquisition of Trigo® and DWL come into play. It is also about the relationship technology acquired from SRD in the form of entity analytics, to an on-going real-time basis look at the relationship between individuals. This creates significant value-add, that today cannot be matched by the competition.

You should also look at the information accelerator layer, that includes such added value as data models, profile models, schemas and policies for specific industries. There are over 400 of these that are available, not just to IBM clients, but also to IBM Business Partners.

Information on demand is really all about overcoming information complexity, and about information enabling your business applications and business processes. It enables you to optimize your information environment through IT services management, and along with that, reduces the hardware cost through autonomic and grid computing capabilities. The net result is that it builds added value for you.

1.2.2 Moving to real-time

So what are some of the changes that moving to a real-time enterprise brings with it. We have summarized some of them in the following list.

- **Event driven:** Rather than relying only on analysts to analyze you data and take an action, processes need to be event driven. The events trigger action, such as the development of real-time business intelligence through the use of real-time analytics. Embedded, or in-line, analytics interact directly with the business processes, triggered by alerts and based, for example, on key performance indicators and business rules. There are also embedded data mining techniques that can be directly invoked, as well as the direct enablement of such capabilities as OLAP and the creation of components such as data cubes for analysis.
- **Rules-based:** To enable event processing, and particularly an automated response to an event action, there is a need for a rules-based environment. These business rules can be accessed by analytic applications, for example, for automated decision-making action. The rules base must be tightly integrated with the business processes, and kept up to date. So, the rules should be flexibly implemented so they can be easily changed or updated as the business requirements change.
- **Updating the data warehouse:** Updates to the data warehouse have historically been performed primarily in a batch mode. That is, the updates are typically performed on a scheduled basis, such as nightly or weekly, with a batch of transactions that have been collected over some period of time.

  The requirement now is to perform those data warehouse updates in more of a continuous streaming mode. That is, updates may need to flow to the data warehouse in more of a continuous manner, soon after the transactions occur. Of course, this will require changes to the business processes and the IT support processes. It will also require changes to the ETL (extract, transform, and load) environment.

For simplicity, you might think of it as processing ETL batches of one record. However, there is more involved, and it may mean also embracing a technology called ELT (extract, load, and transform). Here the update transactions are all loaded directly into the data warehouse before any transformations occur. This enables a significantly faster load process (with ETL processing, updates are transformed and loaded a record-at-a-time). Once the updates are all loaded, then the transformations can be performed, using set-processing. The result can be a more highly performant update process.

You may also need to modify your enterprise application integration environment to support direct integration with the data warehouse. The data warehouse is moving beyond the capability to simply store the data.

- **Separation of the data warehouse and the operational systems:** Historically, the data warehouse and operational systems have been totally separate environments. With real-time, we will have mixed workload systems where both the data warehouse and operational systems may now coexist in the same data warehousing environment.

  This is particularly the case with an operational data store (ODS). What was once primarily thought of as a staging area for updates to the data warehouse, is now taking a more active role in data warehousing itself. That is, queries now access both the data warehouse and the ODS directly to provide a more current (closer to real-time) set of data.

- **Self-contained historical data warehouse:** The data warehouse, as originally conceived, was itself an environment. Queries only accessed the data warehouse to satisfy user requirements. However, there are many conditions where additional data is required to satisfy the query. And, it is not always practical or necessary to copy that data into the data warehouse. The answer is to create a federated environment consisting of the data warehouse, along with other data sources.

  Those data sources can then remain in place, rather than being copied into the data warehouse. The data is accessed, and combined with data from the data warehouse, to provide a more detailed and current set of data. With IBM capabilities (such as WebSphere Information Integration) those sources, even remote and heterogeneous ones, can all be accessed as though they were all on a local DB2 database server. This enables a consolidated multi-level view of the information in the enterprise.

- **Latency in development and deployment of BI applications:** The data warehouse is evolving. It has historically been used primarily as a static data source used for strategic analysis and decision-making. In that environment latency, and the speed of response, were not significant issues. Now they are. The data warehouse is becoming an environment that can also be used for more tactical, and operational system oriented, analysis and decision-making.

For example, it can be used for real-time proactive problem avoidance, and management of business performance. As such there is a need for fast response times to user queries, support of embedded analytic applications, alert processing, and closed-loop problem detection and resolution. And the speed of deployment of these BI applications is critical.

- **Portfolio of standalone tools:** The tools environment is also becoming integrated. Having many tools that operate independently is changing to tools that are part of an integrated BI platform. For example, they may all have the same user interface (or operate through the same client interface), and automatically communicate and pass data to each other to satisfy the user requirements.

Changing technology

As business and technology continue to change and improve, a new phenomena is occurring. The two, originally opposite, ends of the business intelligence tactical and strategic decision-making spectrum are moving together. This is enabling the acceleration towards what is called *closed-loop feedback*. That means the results of data warehousing, or business intelligence, analytics are being fed back to the operational environment. Events can now be acted upon almost immediately, avoiding costly problems associated with delay. This is a significant leap forward, and can now be realized with the capability to support real-time.

The Internet is also playing a key role in this movement. For example, investors now have access to information as never before. And, they can move their money in and out of investments very quickly. If one company is not performing, they can quickly and easily move their investment to another. Now everyone seems to have the need for speed. They want results, and they want them now! They want information, and they want it in right-time!

The move to the on demand real-time enterprise brings with it the opportunity for a proactive, closed-loop environment. That is, as business events happen, they are captured and moved into either the operational data store (ODS) or the enterprise data warehouse. With effective monitoring, and the use of key performance indicators, out of threshold situations can be automatically detected and alerts generated to prompt corrective action. Or, the out of threshold situations can be manually detected via data analysis. If the situation requires corrective action, that action can be dictated either manually, or by the rules base and subsequent execution of an analytic application - as examples. This event-action process effectively provides what we call a *closed-loop, or sense and respond,* environment.

Having a proactive, closed-loop environment takes us another leap forward in efficiency and effectiveness. A more detailed view is depicted in Figure 1-13. Briefly, you can see that the environment:

- Is highly integrated
- Enables continuous flow of data into the enterprise data warehouse
- Is event driven
- Has automated action responses
- Is service oriented
- Uses services provided by WebSphere
- Uses management dashboards for easy monitoring
- Posts alerts for out of tolerance conditions
- Uses alerts as a call to action
- Uses analytic applications to automatically respond to alerts.

This is an example of the environment in action, and represents an environment for which you should be striving. It is the primary subject matter of this redbook. We provide more detailed discussions of this environment in Chapter 2.

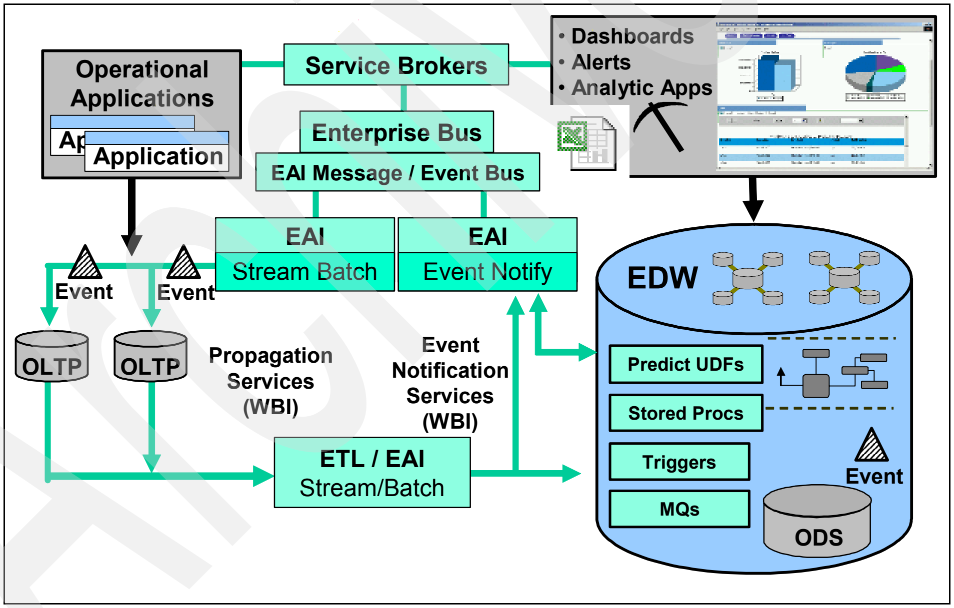


Figure 1-13 On demand real-time enterprise in action

Change, flexibility, and speed are now key requirements for business survival. And moving forward towards the goal of the on demand real-time environment is vital to business leadership and business advantage.

IBM has a strategy, and the solution components, to enable you to get there. Are you ready?

1.3 Contents abstract

In this section, we give a brief description of the topics presented in this IBM Redbook and how they are organized. The information presented includes, as examples, an overview of real-time, data warehousing, information integration, and business intelligence. The focus, however, is on the combination of those subjects. More precisely, it is on techniques for developing a real-time enterprise that can enable real-time business intelligence.

We know our readers will be from varied backgrounds, have varied levels of interest, and typically limited reading time. The abstracts provide enough information to enable selection of those topics that are of most importance to you. We hope you will find them all of a sufficient level of interest that you will be enticed to read the entire book.

We start with a brief overview of the IBM Redbook contents, by chapter:

- Chapter 1- This introduction: Here we provide a high-level overview of the real-time enterprise. It describes and positions terms such as real-time and on demand, and summarizes some of the requirements and changes realized in developing a real-time enterprise.
- Chapter 2 - The real-time enterprise: In this chapter we take a closer look at the meaning of the real-time enterprise, address some of the technical challenges for implementing such an environment, and then consider the solution capabilities that IBM has to offer:
- Chapter 3 - Architectures for the real-time enterprise: In this chapter we describe some key considerations, and some of the key component functionality required, as you begin developing an architecture to support your real-time enterprise. We also discuss the demands for the various infrastructure technologies to support the real-time enterprise that are available today. And lastly we look at some future trends and directions of the initiative, and of the next generation of infrastructure components.
- Chapter 4 - Industry specific real-time enterprise examples: Here we discuss businesses in a number of industry segments, their requirements for real-time business intelligence, and their evolution towards becoming a real-time enterprise.
- Chapter 5 - IBM technologies supporting the real-time enterprise: In this chapter we introduce you to several IBM technologies and products that can be used when developing your real-time enterprise. A number of these

products were used in this redbook project testing environment. We discuss how they interact, and their role in a real-time enterprise environment.

- Chapter 6 - Real-time data warehouse update approaches: Are you ready for real-time? This chapter describes a number of approaches that can be used, depending on your particular requirements and your current environment. It describes a number of techniques and technologies available to help as you move forward with your real-time enterprise.
- Chapter 7 - The project test environment: What did we have as a test environment while developing this redbook? In this chapter, we describe the software and hardware used, and give you a macro view of how those products relate. We investigated two approaches, based on two separate versions of WebSphere Business Integration. New and powerful capabilities made the task much easier.
- Chapter 8 - A retail industry case study: To validate the techniques and technologies discussed in this redbook, we decided to demonstrate a case study. It revolves around a retailer and the processes used to discover, analyze, and resolve a business problem. It this case, the handling of a defective product, provided by a particular supplier.

That is a very brief abstract of the redbook contents. We think it will help in your selection and prioritization for reading topics. But of course, we believe you should read the entire book and that you will find it both interesting and informative.

2

The real-time enterprise

In this chapter we take a closer look at a description of the real-time enterprise, address some of the technical challenges for implementing such an environment, and then consider the solution capabilities that IBM has to offer.

From one perspective, part of a real-time enterprise is related to information technology and part to the business itself. For example, in a real-time enterprise, transactions and data, in many cases, are flowing in more of a continuous manner rather than only being processed in batches. And, where required, the direction is to design and manage the business processes to be flexible and capable of supporting such a continuous flow environment.

The flow of data from the process steps should tend towards continuous. Where required, and justifiable, the data should flow from the process step reporting system to the data warehousing environment (EDW), where it will be available for analysis and decision-making. An alternative approach that will also be covered in this redbook is the use of the operational data store (ODS). With this approach, the data first flows into the ODS, and then, at the appropriate time, into the EDW.

Continuous flow means that the time delay from completion of each process step to storage of the data in the EDW, should tend towards a zero latency - where required and justifiable. This real-time data flow enables the most current data to be available for analysis and decision-making. With this more current data, management can more proactively manage the business processes, to meet the business goals and measurement objectives. And this is where we will see a significant business advantage.

At this point in time it is likely a fairly safe statement to say there are few, or no, enterprises that can claim to have real-time data flow throughout their enterprise, or even in all those business areas that might require it.

That brings us again to a point of contention, because it also depends on your particular definition of a real-time enterprise. We provided our *working description* of that term in "Real-time and the enterprise" on page 5. But we also conceded that such a definition is always relative to critical information needs of the organization, and your particular definition of real-time. If you feel that real-time means no time lag, or zero process latency, then we are pretty safe in our statement.

We also know that any process, or change to a process, requires some time lag. Even if it is only the lag in time from the completion of the process, or task, to the reporting of that completion. That is why our working description of the real-time enterprise takes us past the reliance simply on latency. We continue developing the description throughout this chapter.

2.1 A bit of history

During the industrial revolution, corporations sought competitive advantage through process improvement and innovation in manufacturing. For example, consider the adoption of the overhead trolleys, then used by the Chicago meat packers, in 1914 by Henry Ford for auto production. That gave the Ford corporation a significant advantage over the competition. From that beginning, came a term that was then used to mean the standardization of products for mass manufacturing. That term was *Fordization*. It was a dramatic and lasting direction that impacted all of manufacturing.

Figure 2-1 illustrates one way to predict the possible fate of a business based on their pursuit of innovation and competitive advantage. It enables them to be positioned simply based on their commitment and aggressiveness for improving efficiency and effectiveness in their business processes. In which quadrant would you place your company, and your competitors?

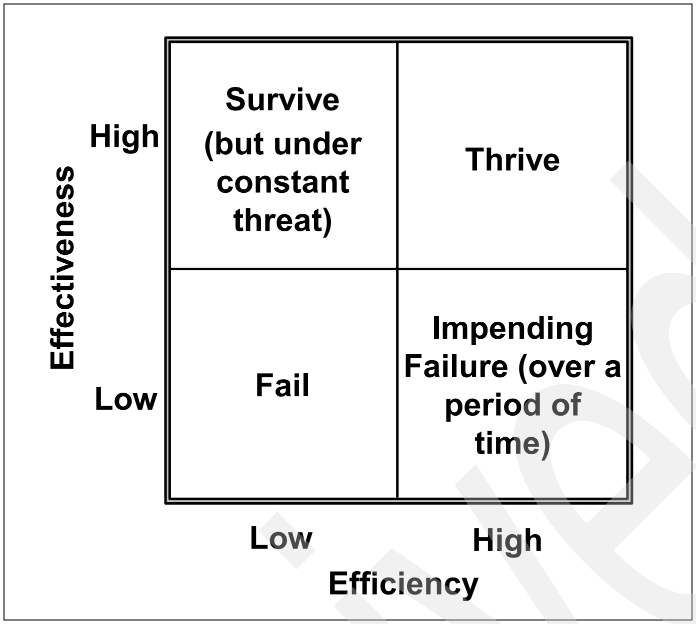


Figure 2-1 Competitive threat

A more current example of innovation might be the use of *Just-in-Time* manufacturing, again, at least initially, in the automobile industry. It is an inventory strategy that was developed to minimize inventory and associated carrying costs. It changed the paradigm from ordering raw materials and parts for inventory, based on some estimate of demand or economic ordering quantity, to ordering for, and at the time of, the actual demand. Companies needed less storage space since they did not have to store all the excess inventories, which resulted in significant savings. And, they did not have to purchase the inventory ahead of time, which also resulted in significant savings.

Just-in-time is enabled by a signaling system called *Kanban*. This is a Japanese word that translates literally to *sign* or *instruction card*. These cards were used to signal when additional inventory was needed in a manufacturing operation. That would then trigger the movement of parts and supplies to the points where they were needed.

By implementing Kanban, a company could move from a centrally controlled supply-push system to demand-pull system. The initial implementation gave Japanese automobile manufacturers a significant cost advantage, and enabled delivery of automobiles based on actual demand. This also resulted in savings to the automobile dealerships in terms of automobile inventory storage space and the associated inventory carrying cost. It was a true paradigm change for the automobile industry.

Functional automation

Organizations have spent years dealing with the realities of functional automation. For example, with the limitation of monolithic applications that only do one thing well, and with ad-hoc integration projects that are expensive, time-consuming, and difficult to maintain. This is primarily due to the lack of standards, a legacy faced by the IT industry. This process of functional automation and innovation is illustrated in Figure 2-2.

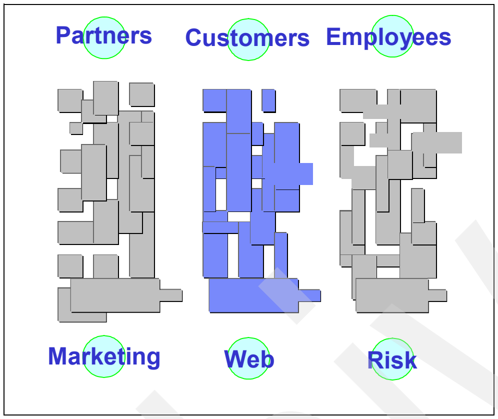


Figure 2-2 Functional automation

Now lets take the functional automation that has gone on around the sales organization, as an example. Most companies have spent huge amounts of money on sales force automation and customer relationship management software. This is regardless of whether those applications were custom built, or purchased as packaged software from an industry software vendor.

However, the automation process did not stop there. Organizations have opted to enable their sales force to work round the clock. They can access their applications at any time, from anywhere, and from a growing number of devices. The infrastructures that have been built to enable it all is astounding. There are the servers, storage, communications infrastructure, and security. Then once it was in place for the direct sales force, it was typically replicated to business partners.

But as we all know, the sales organization does not operate in a vacuum. It is part of an ecosystem that includes marketing, and a whole array of Web-based marketing and sales initiatives. The individual disciplines within that ecosystem have undergone similar optimization and automation initiatives. Web-based programs have been created to help generate leads, and to enable customers to access information and educate themselves. We have probably all invested in loyalty programs designed to simplify transacting business with, and keeping the business of, our key customers.

There is also a burgeoning set of applications and corresponding infrastructure around commerce, lead tracking, digital asset management, and the holy grail of personalization. And, now there have been advances to make horizontal process integration possible.

Horizontal process integration

The fact of the matter is that organizations have spent huge sums of money optimizing the discrete functions of sales and marketing, and there is questionable gain left in wringing more efficiency from either process. The real gains are in the gaps between the disciplines. Creating joint processes backed by information-rich records on every prospect and customer can significantly improve make the marketing and sales operation significantly better.

The breakthrough will come when it is possible for businesses to self-identify, or to tell you what they need and how soon they might need it. And to have a marketing infrastructure that can nurture a customer individually, and proactively. That is, it can provide the right information at the right time. It can do this by recording every tactic so that when sales close, marketing will know what information was used. This then enables them to evaluate and optimize their spend.

But to create that kind of integrated horizontal process, you will need discrete bits of process and application functionality that exists within the confines of sales, marketing, and the Web. This is depicted in Figure 2-3.

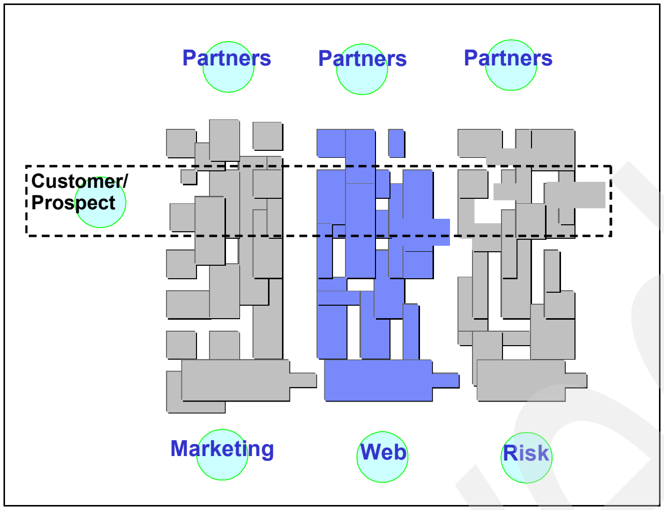


Figure 2-3 Horizontal process integration

This is possible today. There are now standards for creating services and enabling them to communicate, that are agreed upon by major vendors. And an infrastructure that supports self-defined, loosely coupled services has emerged. Along with that are tools to incorporate existing assets, and for the automation and virtualization of systems resources.

On demand operating environment

We are on the threshold of a new business model, fueled by the rise of the real-time enterprise. We call this model the *on demand* business, and its advantages are enormous. At its core, on demand enterprises respond dynamically to whatever business challenges arise. They will be almost intuitive in their responsiveness to changes in demand, supply, pricing, labor, capital markets, and customer needs. These companies will uncover hidden value as the cost savings derived from streamlining operations can be reapplied to new initiatives. At IBM, we think of on demand as an exciting journey, to be tackled in stages. And, there are benefits to be gained at every stage.

The strategy to make this happen is to build an on demand operating environment, as the example depicted in Figure 2-4. Here we are primarily concerned with:

- Integration: Business flexibility through closed-loop integration of people, processes and information within the data warehouse.
- Infrastructure management: IT simplification through layered data warehouse design for consolidation and logical view of data marts.

The focus is on optimizing the operating environment along two very key fronts:

1. The first is increasing business flexibility through capabilities designed to simplify integration. The ability to connect people, processes and information in a way that allows businesses to become more flexible to the dynamics of the markets, customers and competitors. Obviously, this is important within the bounds of any enterprise and will become increasingly important as companies extend their value nets – and more tightly integrate partners, suppliers and customers into their processes.
2. The second is simplifying the IT infrastructure. This is done by creating an infrastructure that is easier to provision and manage through automation and a single, consolidated, logical view of, and access to, all the available resources.

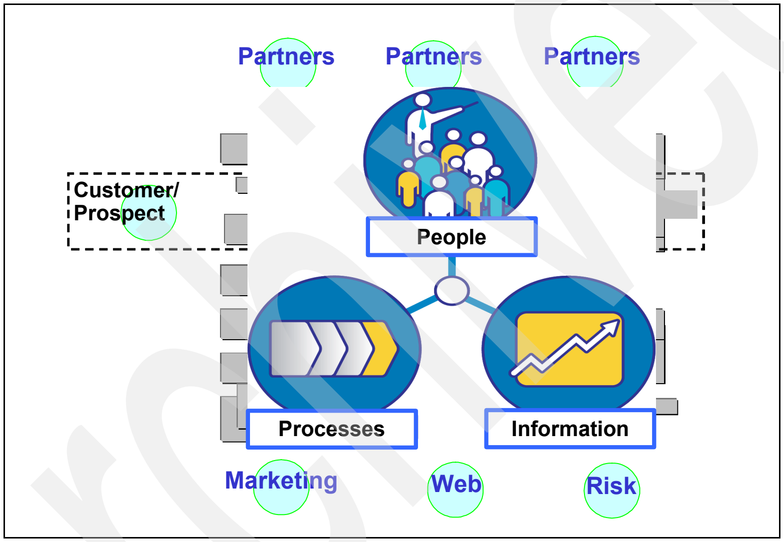


Figure 2-4 On demand operating environment

In some IT organizations, this may seem like heresy. This may be because some organizations have become complacent about the practice of over-provisioning. That is, buying excess capacity to handle the occasional spikes that almost every system is forced to handle.

A major point here is it would be very difficult to build an on demand operating environment without moving to open standards, and a standards-based approach to delivering application functionality and the underlying infrastructure.

2.2 In search of the real-time enterprise

Research for this redbook has shown that the positions and opinions of scholars and business consultants regarding the *real-time enterprise,* and where we are today, is very diverse. However, what justifies the further elaboration on the real-time enterprise is the environment in which many businesses now find themselves.

There is, in fact, a great deal of data and information in any enterprise. It has been gathered over time from numerous sources and stored on numerous targets. That is, most enterprises typically:

- Maintain several data environments
- Use several data management technologies
- Have a number of application development environments
- Store data in silos that exists in separate business functional areas
- Have no consistent programming interfaces to their multiple data sources
- Have few or no IT standards across the enterprise
- Have incomplete and un-maintained business processes
- Use several operating environments and programming languages
- Have data redundancy and inconsistency across the enterprise
- Need to be more proactive making business decisions
- Have several disparate financial and ERP systems
- Have information that is not very easy to understand or use
- Have information not tailored for their use
- Have data that is not integrated
- Take too much time searching for relevant information
- Spend too much of the IT budget on integration

There are significant benefits to the real-time enterprise. But to realize them, many of these conditions will need to be addressed.

The boundaries of the real-time enterprise

Depending on the context, the expression *real-time* can have many different meanings. Quite often real-time information processing is misunderstood to be the same as online processing. It is also confused with high performance computing. Or it is thought to be the time lag between when a query entered the system and the result set is provided to the end-user. Actually, that is simply the response time for the query.

In business terms real-time might mean that the processing time for an operation is shorter than, or equal to, the maximum time lag allowed per the requirement. In the context of real-time, it also seems applicable to mention the term *synchronicity* as the coordination of processes with respect to time. Does that mean that synchronous operations or processes are real-time, while asynchronous processes are not? Not at all. We are not talking simply about speed, but about the currency status of the data. Is it the most current result of an execution of a particular process?

A key in determining real-time data status, is the time lag or delay within and between the knowledge creation process, and reaction time for decision-making and adaptation. Here are a few more attributes that need consideration in a real-time enterprise:

1. Decision makers and executives need to become time-based. To support this requires, as an example, event-based technology. They must be made aware of critical events as they occur, receiving appropriate information to support decision-making.
2. Processes must cross organization and company boundaries. There must be increased emphasis on processes, as they are the vehicle for tracking activity execution and capturing real-time status information. This information should be available to all who need it, immediately. It can be available as an entity, or integrated with other data from data warehouse to be in context.
3. Business process changes must occur quickly (for example, in a matter of hours rather than days). Processes should be integrated into the overall information infrastructure. As such, they participate in the delivery of real-time data. And, to do so, they must be kept up-to-date and accurate. Moving to an automated process development environment can make this update requirement easier and faster to meet.

2.2.1 Attributes of the real-time enterprise

We have discussed the terms real-time and on demand. They have meanings themselves. But when we put them together and form the term *on demand real-time enterprise*, we have a new entity. Unfortunately it is an entity that can also have many different definitions. So, we will not try to define it. But, rather we will characterize it. What are some of the characteristics of the on demand real-time enterprise? Or, what are the attributes by which we can recognize such an entity?

In the following sections, we discuss and describe some of these attributes.

Agile and flexible

By definition, an environment that is agile and flexible is able to deal with change. This can take several forms. For example, integrating applications and the data warehouse environment with processes for access to real-time data. The enterprise needs to become *process aware*. That is, to move towards a flexible environment where processes, and process workflows, can be changed in real-time, based on process feedback. This can, as an example, be implemented by using processes based on reusable and easily replaceable services.

Information infrastructure

To satisfy the demands of a real-time enterprise requires a well designed and robust information infrastructure. That consists of well designed data models, robust database management systems, applications, and data stores.

To get there can represent a change, or evolution, over time. It is a movement towards new technologies. For example those that support loosely coupled systems and applications, and re-usable components (or services). They should be based on standards for openness and flexibility. For example, such services should be build on standards such as a services oriented architecture (SOA).

For communications is this loosely coupled environment, consider open standards that can enable flexibility in any-to-any communications. This type of communications can be provided, for example, by the WebSphere enterprise service bus. The WebSphere suite of products offer solutions that can help provide you with such a robust information infrastructure.

Data warehousing

The data warehouse is a key component of the enterprise information infrastructure. Initially built to house history information, it now houses more and more real-time, or near real-time, data. This requires some changes to the IT architecture that was designed to house history data for use in strategic decision-making. Now the data warehouse is required for use in strategic and tactical decision making. But to do so, the data must be closer to real-time.

Historically, applications were built or bought to transform the data in the operational systems into a format suitable for loading in the data warehouse. This transformation and loading was typically performed in a batch environment. The batch cycle may have been measured in hours, days, or even weeks, but now that cycle time must be decreased. We have listed, as examples, three alternatives to satisfy the data transformation and loading requirement.

1. Extract, transform, and load (ETL): Here the data is extracted from the operational system to an ETL server. It is transformed per the requirements of the data warehouse design, and staged for subsequent batch loading into the data warehouse. Here the transformed data on the ETL server must be transferred and loaded a record at a time into the data warehouse. This is a simple and efficient process, and is still suitable for a high percentage of data warehouse updates. In particular, those that are not required for real-time activities.
2. Extract, load, and transform (ELT): This process was developed to enable faster updating of the data warehouse. Here the data is extracted from the

operational system and transferred directly into the data warehouse. The subsequent transformations can be performed quickly once in the data warehouse. This is because the transformation application can use set processing, rather than record-at-a-time processing. Though still somewhat of a batch operation, this can result in a performance improvement in the time to update the data warehouse.

3. Continuous update: This technique requires more change to existing systems, particularly those implemented with batch oriented ETL processing. With this technique, applications take the operational transactions, transform them, and put them either in staging tables, or in queues such as WebSphere message queues. The data is then automatically taken from the queue and processed against the data warehouse. This type of approach is depicted in Figure 2-5.

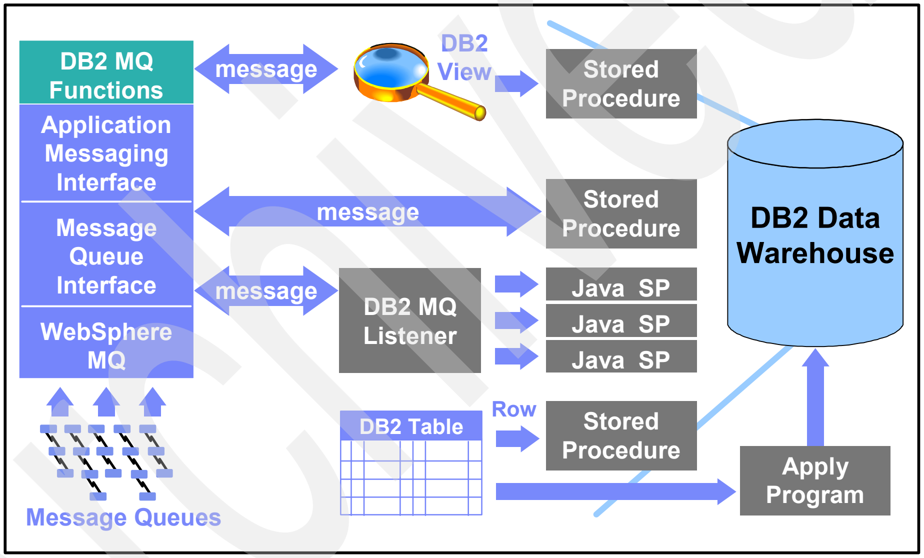


Figure 2-5 Approaches for continuous update

This continuous process can keep the data in the data warehouse updated in near real-time, and can be a very effective approach. As you can see in Figure 2-5, there are several alternatives for implementing this approach.

For more information about this particular topic, see the IBM Redbook “Preparing for DB2 Near Realtime Business Intelligence, SG24-6071.” And, we also explore this subject further in Chapter 5., “More real-time enterprise enablers” on page 223. However, this can bring with it a good deal of expense and processing overhead.

The question can now be asked, is all this necessary? Is it justified? You will need to research that, and then decide for yourself. But, what, you ask, is the alternative? Well, it may not really be an exact alternative, but there is another approach for making real-time data available in the real-time enterprise.

That approach is to maintain many of the batch oriented processes that are already in place, performing adequately, and are relatively inexpensive and simple to use. Then focus on implementing real-time in those areas where it is required and justified.

Another approach for satisfying the need for real-time data is to use the ODS and a data federation approach with the enterprise data warehouse to satisfy the requirement. What does that mean? It simply means that the ODS now takes an active role in the data warehousing environment.

That is depicted in Figure 2-6. You can see that the ODS is placed inside the EDW, and participates, for example, in supplying information that is used for the real-time status information and analytics that gets displayed on the dashboard.

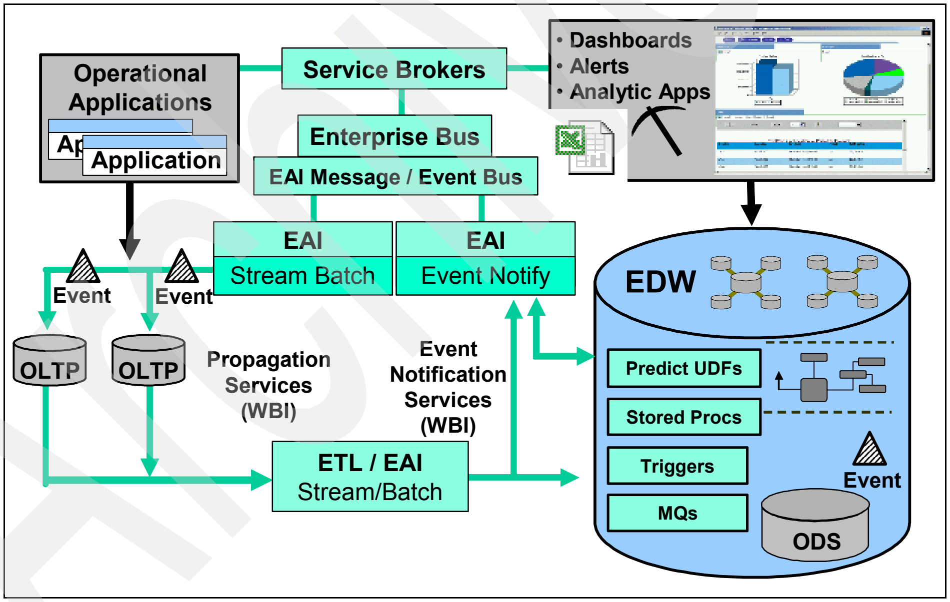


Figure 2-6 On demand real-time enterprise in action

The ODS is still used as a source for updating the data warehouse, but without the requirement to update it in real-time.

Data mart consolidation

Data marts can bring with them many benefits for the enterprise. However, care must be taken to maintain control. When control is not maintained, data marts can proliferate at a rapid rate. Along with that, comes some of the downside of having data marts. As examples, significant data redundancy, additional servers, data management systems, and maintenance costs. To say nothing of the impact of having the same data in different data marts at different levels of currency. This can lead to inconsistency and inaccuracy in the information across the enterprise.

Part of the issue is the existence of independent data marts. These are data marts that are not sourced and managed relative to the enterprise data warehouse. Because of this there is an even greater risk of inconsistent and inaccurate data in the enterprise.

Dependent data marts are sourced from, and managed by, the enterprise data warehouse. So, there is less risk with the data. But, they do come with a requirement for additional hardware, software, and maintenance. This additional cost needs to be justified.

Because of data mart proliferation, many companies are now taking a closer look at the requirement for data marts. And particularly at independent data marts. From this has come an industry-wide initiative for data mart consolidation. There are significant savings involved, as well as the elimination of the associated risks previously described. For more information about this topic, see the IBM Redbook “Data Mart Consolidation: Getting Control of Your Enterprise Information, SG24-6653.”

Integrating and automating the business processes

The enterprise needs to become more *process aware.* This requires a move towards a flexible environment where processes, and process workflows, can be changed in real-time, based on process feedback. This will be based on reusable and easily replaceable services.

This is more than simply automating processes, although that is definitely a recommendation. It is the integration that is important, and the development of processes with standards and services to make them more easily modifiable. This is part of the loosely coupled application development philosophy we described in “Information infrastructure” on page 38.

Having standards to support this loosely coupled environment is key. For example, standards to support communications among and between the various resources. For this, IBM again has a solution. That is a common information bus, called the enterprise service bus. That bus is instantiated in the WebSphere

Message Broker, V5, and is called the Advanced Enterprise Service Bus. In Version 6, it became the WebSphere Enterprise Service Bus.

A key component in automating the processes, is to then monitor them - automatically. To make the monitoring effective for managing the process you need guidelines or thresholds that let you know when things are not going so well. These are called key performance indicators. Values are set, with specific threshold boundaries, and then the process is monitored. If a threshold set for the process is exceeded, then an alert must be posted, to call for action. It may be an automated action, or it may require human intervention.

It is through the monitoring of the processes that we are enabling their management. It is a proactive approach to avoiding problems rather than simply taking action after they are discovered to minimize their impact.

Analytics

We have primarily been focusing so far on data and getting it in to the enterprise data warehouse. The primary reason for getting the data in to the data warehouse is to transform it and organize it to make it easy to access and use - primarily for decision-making.

Now we focus on getting the data out, and using it. There are several approaches here, depending on the requirement. For example, data analysis and reporting is probably the best known and most widely used approach. But as we move towards the real-time enterprise, there are additional techniques that need to be considered. As examples:

- Analytic applications: These are applications written to perform data analysis for the user. This enables fast and consistently repeatable decision-making, because it is based on rules, rather than human decision-making. These may also referred to as *inline analytics*.

  As usual, such capability brings with it other requirements. For example, to enable the application to make decisions there needs to be a set of rules for it to follow. This set of rules must be frequently audited and maintained to assure the are still applicable and correct.

- Guided analysis: This is an approach to enable decision-making by people who may not be as technical or as close to the process they are analyzing. If more people could analyze data and resolve issues, it would significantly raise the efficiency and effectiveness of the enterprise. It can be thought of as an analytic application that works with issues that cannot be resolved with a strict rules base. Some analysis, logic, and decision-making by a human must be involved.

  This is accomplished by capturing the knowledge and decision-making process of a knowledgeable data analyst, and making it available to others through an automated means. For example, the application may come to a point where there is not just one potential solution, but there are alternatives to be considered. The application knows what they are and can provide that information the analyst, but the analyst must investigate each and make a decision as to which applies. Although not totally automated, this approach also enables significant improvements in efficiency and effectiveness for the issue resolution task.

Dashboards

Dashboards are becoming the window to, and control panel for, the enterprise. Rather than getting static reports, that are typically already out of date, dashboards provide a dynamic information mechanism. That is, the information is dynamically, or updated in real-time. This enables monitoring the status of activities in the enterprise to prevent them from getting out of control, enabling real management of the processes and the enterprise.

To help monitor, we have the key performance indicators. When a set threshold has been reached, an alert is posted. That is the indication that action must be taken to avoid a problem. That action could be automated, or require human intervention.

Closed-loop feedback

When the issue is resolved, there needs to be feedback to the process. As examples, what was the issue and what was the resolution. And, a recommendation for a change to the process that would avoid that issue in the future, if there is one. And then action may be taken to change the process.

This is referred to as closed-loop feedback. That feedback, and subsequent action, can be performed either manually or as an automated task.

One of the key recipients of real-time data, and a key component of a real-time enterprise, is the business intelligence environment. Lets take a look at that environment.

2.3 Positioning business intelligence

A primary reason to create a real-time enterprise in the first place, is for business intelligence. That is described as intelligence to enable proactive management of the business. For example, proactively managing activities to meet the business goals and performance measurements. It lets you really manage the business to avoid problems, rather than managing problem resolution. And the key ingredient to enable all this is data. Do you understand your data? Do you know where it comes from, the exact meaning of it, where you can use it, and when you can use it? Lets first take a look at data.

Here we present a high-level view, a logical view, of the data. We call this the IBM layered data architecture (or information pyramid), and it is depicted in Figure 2-7. It gives us a logical view of how we can build an enterprise data repository that is enabled by analytics and other key business applications. The key is to leverage an enterprise set of data, and build in a performance layer to enable analytics.

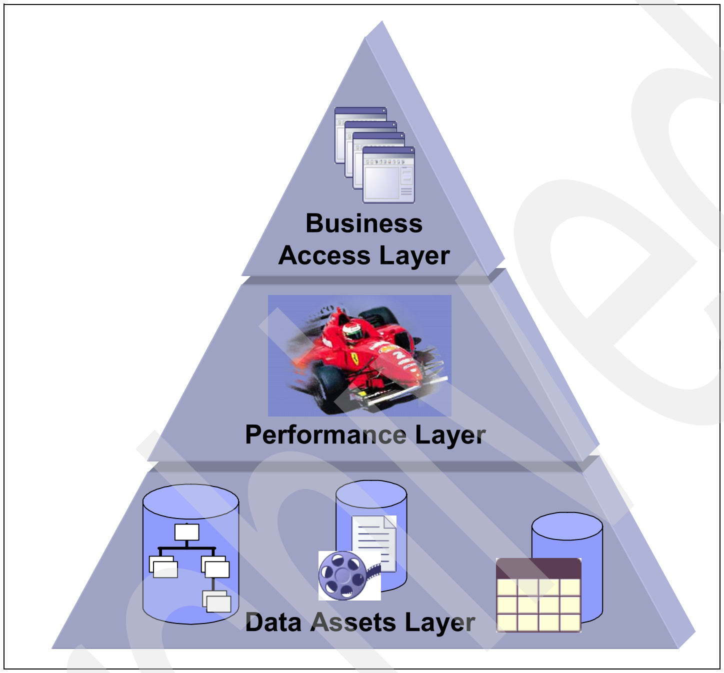


Figure 2-7 IBM layered data architecture

The layered data architecture includes:

- Data Assets Layer: These are the data assets of the enterprise, consisting of an application neutral data model (typically in third normal form), structured and unstructured data, and the results of enterprise application integration and enterprise information integration. It is the atomic layer of data that provides the foundation, information infrastructure, of the enterprise data asset.
- Performance Layer: Here are the components and capabilities to enable optimized performance, typically by means of auxiliary data structures, such as MQTs, indices, and application specific tables. They enable the capability for power analytics, data mining, and OLAP.

- Business Access Layer: This is the window into the enterprise data that enables analysis and the conversion of the data into actionable information for business decision-making.

Now here is a different view of the layered architecture. This view shows the same logical architecture but overlaid with business data and indicating where that may fall in the layered approach. We refer to this as the layered data warehouse architecture, and is depicted in Figure 2-8.

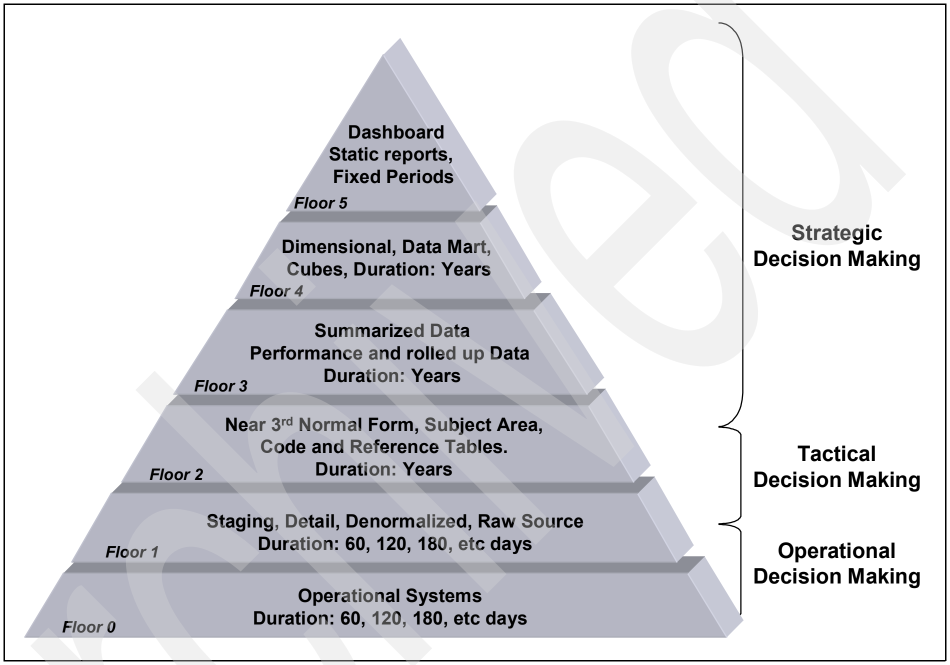


Figure 2-8 Layered data warehouse architecture

The idea here is to map the delivery of data needs to your applications and organization needs. We call this *right time data delivery*. As you can see, the lower floors of the architecture offer the freshest and most detailed data to the enterprise. Here is where you will make operational and tactical day to day decisions. As you move up the floors, the data becomes more summarized and focused for specific users or applications. In floors 3, 4, 5 you will find users doing more strategic analysis.

Our combined IBM solution will allow you to physically support a design that is tailored to the needs of your business—giving your users access in real-time to point-of- sale (POS) and inventory data while at the same time providing summarized views to re-occurring business questions such as, what are the sales in this region, how do my stores compare in performance, and who are my top customers?

This design, when mapped to your enterprise, will help determine the right number and type of building blocks you need to build your data warehouse. The building blocks will be determined by such things as the number of users, complexity of workloads and how active your warehouse is in terms of data delivery.

Real-time enterprise and BI

BI in the real-time enterprise is enabled by three primary capabilities:

1. ETL and ELT processes to populate the enterprise data warehouse (EDW). And, data federation to enable the inclusion of heterogeneous sources of data. The operational data store (ODS) is part of the EDW environment, and provides the most current data to enable real-time BI.
2. an RDBMS that can support continuous data loading without disrupting the query environment. Loading must be able to continue along with a continuous and concurrent query workload. And, it must maintain acceptable performance in both environments.
3. personalized Web pages and portals for more traditional data access and analysis. The real-time enterprise has two primary capabilities for BI data delivery. These are static and interactive dashboards, to support data subscription, online process monitoring, and manual and automated action for advanced data analysis and problem resolution.

Right-time information

So far, we have primarily been describing the real-time enterprise, and real-time data. We needed to do this to get a common understanding of those terms in order to decide how they apply in general.

Now we have another term that seeks to extend, or further clarify, these terms. That new term is *right-time information, or right-time BI*. That is a term that has more of a focus on BI, rather than all the workings of an enterprise and the extended business partners. After all, data is not very valuable until it is put into some context. When that happens, it becomes information. Or, as we like to say, *actionable information*.

For more on this topic, we have taken some information from an IBM article called “Right-time information - right now. How to turn the data you get into the information you need” and included it in the following sections. This article is very pertinent to the topic of this redbook. You can determine how to access the entire article by referring to “Online resources” on page 361.

Right-time information right now

Strikes. Materials shortages. Backlogs. Opportunities do not wait—and neither do crises. That is why businesses need to know what is going on, not just in their organizations, but in the companies they buy from, sell to and partner with. In a world governed by demand, your business needs to get it right the first time. And for that, you do not just need real-time data — you need right-time information.

Real-time data and right-time information sound similar, but there is a critical difference: Real-time data is the 27 days your main supplier is backlogged. Right-time information is a warning that, if the backlogs continue and labor and facility costs remain constant as a result of an unchanged production mix, overall output will drop by 10%, and revenues by 40%.

Maybe your IT department is already feeding you real-time data about your finances, supply chain, partners and personnel. Perhaps you have a team of analysts that monitor key business processes and provide information about pressure points within the organization. But can those analysts:

- Provide objective measures for each division against driving the corporate strategy?
- Unify different sales teams around a single goal?
- Grade your Orlando plant manager's labor plan against the regional growth strategy on a daily, weekly or monthly basis?

With right-time information driven throughout your enterprise through the judicious use of business performance management (BPM) balanced scorecards, the answer to these questions can be a resounding "*yes*"!

Step into a right-time world

Many mistake real-time data for right-time information. They think the people in the IT trenches and marketing departments will scrub the data and come up with the gems that executives can use to guide the enterprise to success. But the truth is that a unifying, effective strategy is the first step to creating right-time information out of real-time data. And strategy is driven from the top.

Real-time data reflects activity without concern for new business initiatives, product launches, or effects from regional expansion or world events. Only corporate leaders have the vision, insight and experience to identify outside occurrences that can make or break a business.

The recommendation to executives is to start with their corporate strategy and then drive down to key business processes—sales channels, production or services—that support the strategy. Using the business processes as a guide, the leaders can then identify key performance indicators (KPIs) that quantify the success of those processes. A KPI can be either real-time data for simple operations, such as daily sales figures, or right-time information, such as book-to-bill in the context of existing operational costs.

Then create a business performance management (BPM) scorecard that drives the responsibility for KPIs to the relevant business divisions. By starting with the corporate strategy, then setting up objective measures through KPIs to monitor the strategy's success, and finally pushing the strategy to the limits of the enterprise through BPM scorecards, each employee can be measured against the over-arching corporate strategy. Thanks to leveraging right-time information, your enterprise is finally unified behind a single goal: success.

Getting started

How can you turn real-time data into right-time information? Here are a few points to get the process started:

- Look around. What information are you able to get from your enterprise? What information do you need? How accessible and how dependable is it?
- Data or information? There is a reason it is called *raw* data. Before you can use it, it needs to be carefully prepared, or filtered, scrubbed, carved into manageable pieces and readied for presentation.
- No fear. Taking a data inventory is great, but do not make the mistake of thinking that existing processes and data structures are the best. Use a top-down approach to determining the ideal form of information for each business process, then design *information supply chain* links to accommodate those needs. It may take more work, but it is better than staying inside your organization's data comfort zone.

The discussion above shows that real-time concepts and right-time concepts are not mutually exclusive but rather go hand in hand. In order to create knowledge or business insight as the base of decision making the data and information provisioning needs to be as fast as possible with minimum latency to be available at *the right time* for the decision maker, analyst, marketing strategist, etcetera.

2.4 Closing the loop

Having a real-time enterprise means more than simply getting data quickly. The reason for the data is to provide information for management decision-making. That is, it is what you do with the data that makes the difference. Having it more quickly gives the capability for faster decision-making.

So, we first need to focus on the two key capabilities we term *getting the data in*, and *getting the data out*. Of course, we mean in and out of the enterprise data warehouse. That is, we need current data available for decision-making, and to have a more accurate view of the business environment. That process is depicted in Figure 2-9.

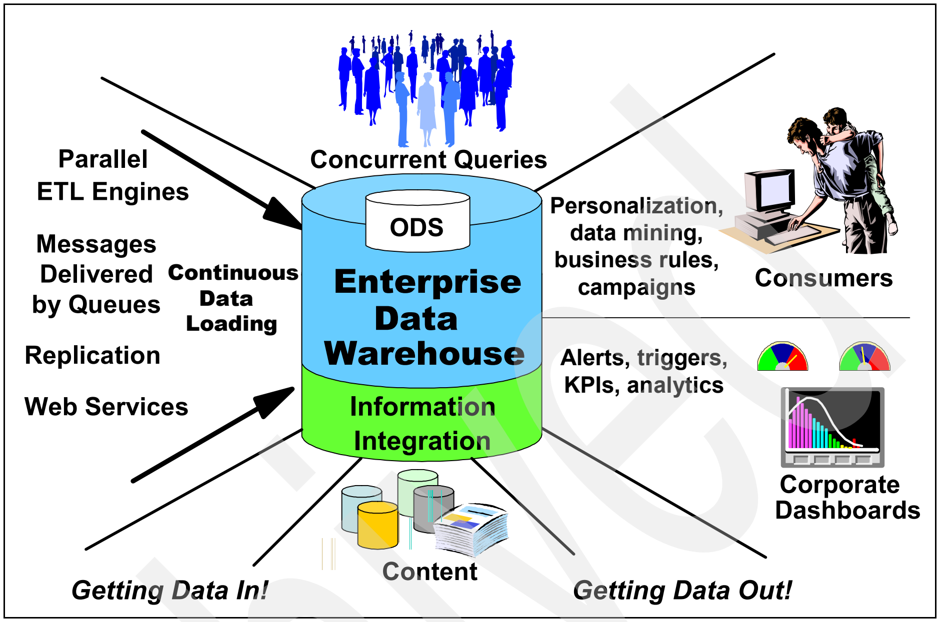


Figure 2-9 Data Warehouse Environment

But once the data in the data warehouse is used for analysis and decision-making, action must be taken. Typically that action is to resolve an impending, or existing, problem. With the action and resolution of the issue, there is a follow-up requirement. That is feedback.

What caused the problem? How was it resolved? How can it be prevented? Is there a change to the process required? Are there new KPIs that should be considered? Are the thresholds properly set for the existing KPIs?

These are a few of the questions that should be asked to enable improvement to the processes and avoidance of similar problems in the future. It is the feedback of the information and the taking action to improve the process, that constitutes *closing the loop*.

Many terms in the business process management area originate in the engineering sciences. It is the same with the concept of closing the loop discussed in this section.

Since we are in the spirit of engineering sciences, take a look at Figure 2-10. Here we have an example of an engineering control loop, which presents an excellent comparison for a closed loop. It does, in fact, show three important components, that also apply to what we call a closed loop.

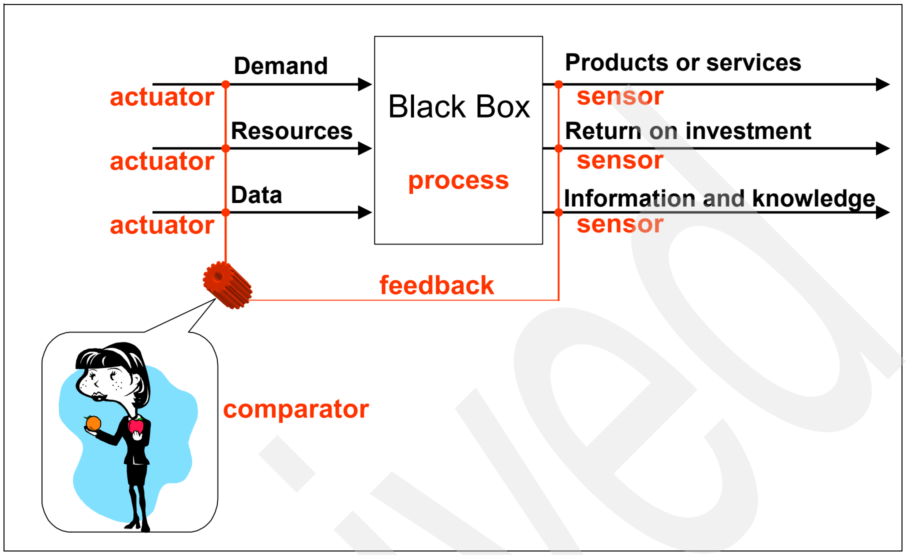


Figure 2-10 Control loop with sensors, comparator and actuator

The three components are:

1. Measurement by a sensor connected to the process
2. Decision in a controller element, or a comparator
3. Action through an output device (*actuator*), for example a control valve.

In the business world the sensor would be part of the business intelligence infrastructure and might trigger an event on the dashboard of a decision maker. The comparator might be the decision maker, analytic application, or an algorithm triggering the actuator to adjust to a new environment.

Closed-loop processing

The following examples from the distribution sector are to illustrate BI and closed-loop process management.

In Figure 2-11 we have an example of collaborative planning, forecasting and replenishment (CPFR).

The Voluntary Interindustry Commerce Standards Association (VICS) is an organization working to improve the efficiency and effectiveness of the entire supply chain. Members are companies who have proven that a timely and accurate flow of product and information between trading partners along the supply change significantly improves their competitive position. A subcommittee of VICS developed a new collaborative dimension to supply chain processes called CPFR.

Figure 2-11 also illustrates the differences of traditional versus CPFR distribution. On the left you see the traditional distribution system which is still very common. Significant amounts of manufactured goods are held in the manufacturers warehouses or in a central distribution depot. Goods are housed in a warehouse in anticipation of demand. As demand occurs, the goods are moved to other distribution centers or retail outlets. Since the store manager cannot always know what will sell in what quantity, they must over stock many items to be sure they do not run out of goods to sell.

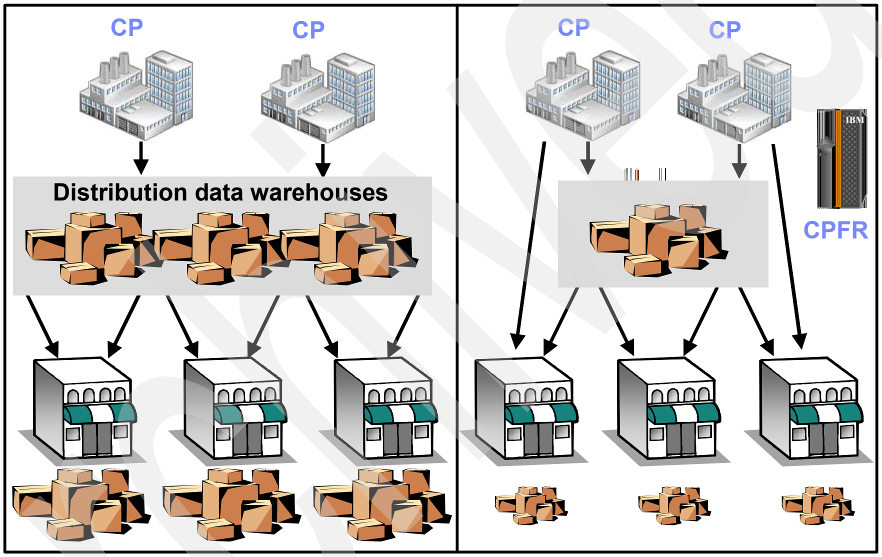


Figure 2-11 Traditional distribution versus CPFR

On the right of Figure 2-11, we replace a majority of the inventory with a computer and a truck. Sales information is shared with the manufacturer and produces precision forecasts of items store by store. The result is the manufacturer can fine-tune their delivery logistics to ensure just enough product is delivered to each store as needed. This allows both the CP and the stores to significantly reduce the inventory in the supply chain. Since your forecasting is much more accurate, you no longer have to build up excessive reserves in the stores or the warehouses. Some inventory still need to be in the locations simply because the truck cannot move instantaneously between, locations.

The objective for CPFR as shown in Figure 2-12 is the integration of supply and demand side processes to reduce cycle time, decreasing fixed asset and working capital, reducing inventory, increasing sales and eliminating charge backs.

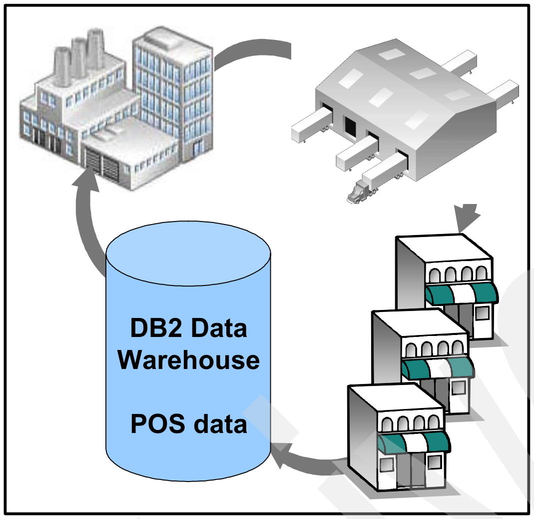


Figure 2-12 The closed-loop in CPFR

Therefore CPFR is closing the information loop between retailer and manufacturer and everybody in between.

Our second example in *real-time* retailing is about price optimization strategies. Here are some aspects of pricing strategies:

- **High-Velocity Item/Competitive Price Analysis**: Helps category managers interpret competitive price data collected by third-party services; on a store-by-store basis, algorithms identify items and categories that are overpriced or underpriced.
- **Medium and Low-Velocity Item/Price Elasticity:** Consumers are slightly less price sensitive when shopping for medium- and low-velocity items; this gives retailers an opportunity selectively to raise the price of these items.
- **Markup/Markdown-Opportunity Identification**: Employ algorithms to review POS movement history; it will identify items with rapidly accelerating (or decelerating) sales volumes, compare the prices of these items to those of competitors.
- **Promotional-Price Analysis:** Setting a competitive price point by local-market average prices versus category manager’s suggested promotional price; promotional sales histories reviewed to understand item movements at various price point.

The example in Figure 2-13 illustrates a *closed-loop process* for a demand-based pricing strategy for a third-party competitive pricing agency. Imagine the agency publishing a fee-based Web-service providing pricing and market information based on a stock keeping unit (SKU) or a product group or even implementing a service using radio frequency identification (RFID) technology.

We see that the pricing, markdown and promotion optimization enables the retailer to quickly evaluate and change prices across all SKUs in response to a real-time demand. This action is based on a comparison of competitive prices regarding other retailers, from a third-party pricing service.

This enables the retailer to set and revise the initial mark-up and prices based on consumer demand, corporate revenue, and inventory level targets. The retailer can then determine optimal markdown timing, depth, and promotional impact for effective yield management.

As Figure 2-13 implies, that series of actions closes the loop on the demand-based pricing activity. This pricing activity is on-going to meet the targets of the retailer for product volumes, inventory levels and turns, revenue, and profit goals.

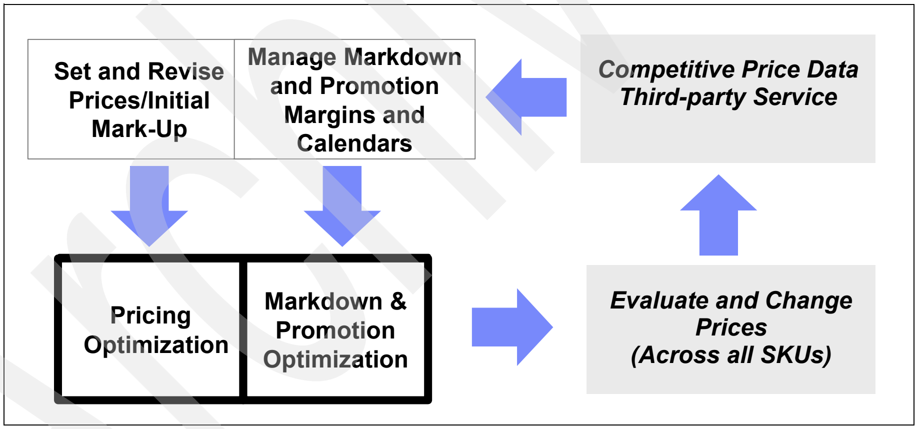


Figure 2-13 Closed-loop process for demand-based pricing

On the application level, Figure 2-14 illustrates how decision making applications interact with transactional applications via standard EAI (enterprise application integration) services such as Web services, JAVA, .NET, or even through transitional Client / Server technology.

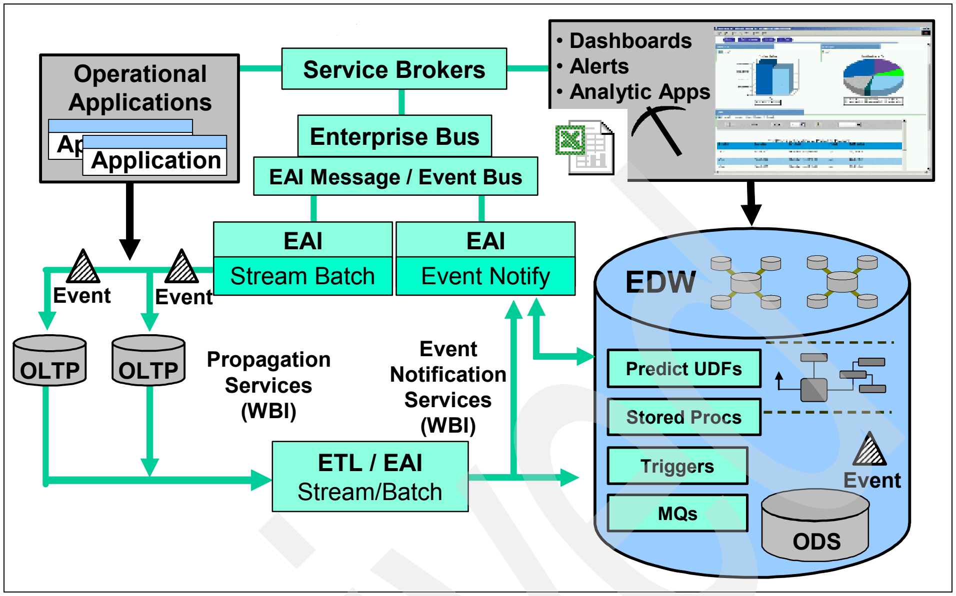


Figure 2-14 Closing the loop between applications

Note the line from the Event Notification and Propagation Services boxes – events can flow to the application for action too, instead of just the operational systems themselves. This is an example of closing the loop at the application level.

2.5 Industry examples

In this section we discuss businesses in a number of industry segments, their requirements for real-time business intelligence, and their evolution towards becoming real-time enterprises.

In general, leveraging low-latency information provides value to these businesses by enabling immediate, and proactive, knowledge based decision-making. We understand that not all areas of all businesses have a requirement for real-time information. But that does not inhibit them from becoming a real-time enterprise. The key is to get started in the evolution.

IBM can help get that evolution moving faster by providing industry specific oriented data models. An example of those data models is depicted in Figure 2-15.

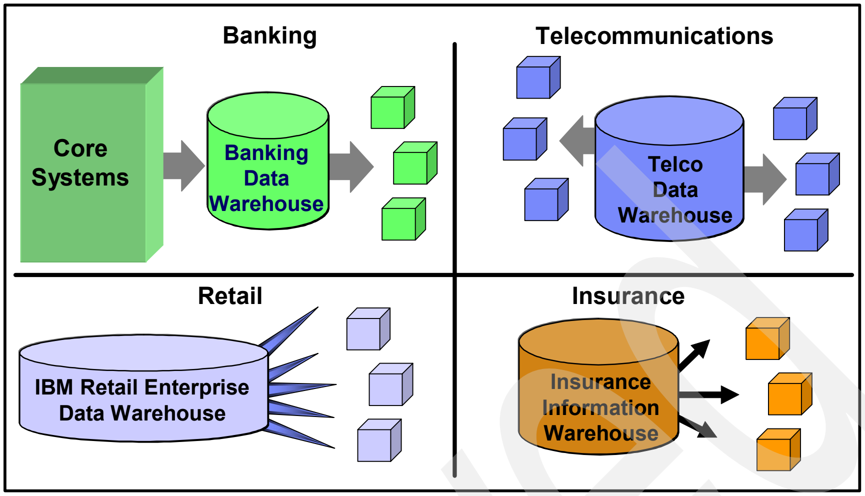


Figure 2-15 IBM Industry Data Models

These models are used as a base for the industry data warehousing environments. The information from the industry core systems are stored in the data warehouse, queried, and used to populate any required data marts.

Banking data model

The core systems in this model are:

- Mortgages
- Demand Deposits
- Investments
- Term Deposits
- Trading
- Credit Cards
- Portfolio Management
- Loans

Examples of banking data marts are:

- Relationship Marketing
- Profitability
- Risk Management
- Asset Management

Telecommunications data model

The core systems in this model are:

– Service Usage
– Provisioning and Network Operations
– Billing, Payments, and Credit metrics
– Customer Hierarchies and Geo demographics
– Customer Segments and Usage Behavior
– Product Hierarchies, Bundling and Rating
– Channel Activities and Commissions
– Campaign Management
– Finance
– External Data

Retail data model

The core systems for this model are:

– Customer Management
– Finance and Payment Systems
– Supply Chain
– Store Operations
– Merchandising
– Point of Sale
– Multi-Channel Product and Service Management

Insurance data model

The core systems for this model are:

– Underwriting Analysis
– Claims and Benefit Analysis
– Intermediary Commission Analysis
– Persistency Analysis
– Predictive Modeling (Fraud Detection and Lapses)

In the following sections, we look at a few industry segments and how they can benefit from evolving to a real-time enterprise environment.

2.5.1 Retail

The retail industry is highly dependent on their supply chain, because if they do not have the goods in stock they cannot sell them. This not only impacts their sales, but also their customer base. If customers cannot find the product they seek, they will go elsewhere. This becomes even more critical as retailers move to a just-in-time inventory model, much the same as has happened in the manufacturing industry. This means that retailers will no longer carry a large reserve inventory on site, and will require good collaboration with their suppliers.

Retailers often highlight specific goods through promotions and sales to attract more customers. In these situations, having the goods in stock that they are promoting is critical to both revenue and customer satisfaction. They also need data that will help them understand the sales potential.

In addition, being integrated with suppliers and shippers, and understanding their capabilities, can enable additional deliveries quickly if sales exceed expectations. A good view, and control, of their goods replenishment systems enables flexibility for handling business fluctuations. By collaborating with their supply chain partners, the retailers can now be more customer oriented and less product oriented.

This customer orientation will require a number of changes in the retail business. As examples, they will need:

- Commercial data: This would include data on passed and planned promotions, pricing factors, and sales forecasts.
- Collaboration: This is consistent with the move to just-in-time inventory model we mentioned earlier. To make this happen requires significant collaboration and integration with the retailer business partners and suppliers.
- Analytics: Here again, information is power. Maintaining a good data warehouse for data analysis provides that information. This includes sales, product, and customer data. It enables optimization of inventories, enlightened pricing, intelligent promotions, and stocking of the best selling products. This information will help management make those decisions in product, inventory, and pricing that will maximize their profit margins.
- Customer relationship management: This will help the retailer look at a different side of the business. For example, it will focus more on product mix, store layout, and related purchasing (buying one product because it is directly or indirectly connected with another). It will also enable more personalization in promotions, advertising, and mailings.

How does all this fit with business intelligence and real-time? To understand that we can look at the interaction of the retail business processes and BI, which is depicted in Figure 2-16. This interaction provides functionality, and increased benefits. As examples:

- Merchandising: Retailers can differentiate their products by delivering more efficient merchandising processes. And, they can provide role based, real-time access to a fully integrated merchandising environment. This will enable them to purchase the right products, at the right time, based upon market demand. This can result in benefits such as increased sales by maximizing in-stock percentages; increased customer satisfaction by on-time order fulfillment; and margin protection by minimizing markdowns.

- Supply chain management: This can eliminate bottle necks and unnecessary costs in the supply chain. Collaboration with the suppliers can enable an automated and integrated set of data, synchronization, and product information management. The costs can be reduced by eliminating manual processes and the errors caused by them, and by trading item information with suppliers. All this can result in accelerated deliveries and reduced Purchase Order errors, reduced stock-outages, and improved collaboration.
- Advertising, Marketing, Promotions: Having the right information enables understanding what are the right products and the right promotions that will attract customers to the store and increase the overall customer traffic. And, as with merchandising, it can enable personalized targeted promotions and improved shopping experiences. This can increase promotion effectiveness, and result in increased sales.
- Distribution and Logistics: As with supply chain management, this lets the retailer focus on reducing inventory expense and out of stock situations. Use of technology such as barcodes and RFID to track individual stock keeping units (SKUs). Coupled with track and trace capabilities, and automated document matching, the retailer can proactively manage the logistics.
- Channel Management: Many retailers want to support multiple sales and fulfillment channels. Channel management can provide a consolidated order platform and a fully integrated set of sales channels. It enable retailers to provide customers a seamless and personalized cross channel buying experience, anytime, and anywhere.
- Channel Operations: This is the area where retailers try to control expenses at the store. It can provide a Web platform that is affordable and manageable, and that allows them to transform the customer and store associate experiences, increase customer satisfaction, and streamline store operating procedures. And it can maximize sales by connecting the in-store applications to provide real-time information.

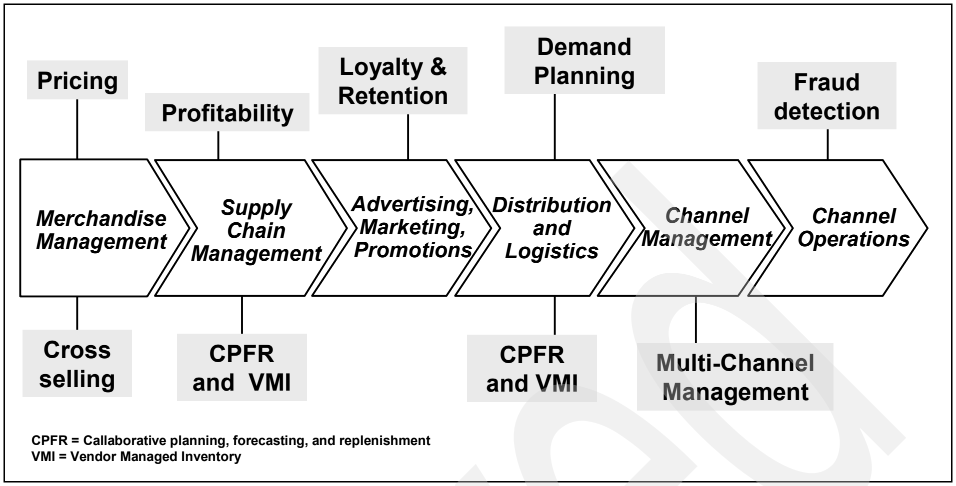


Figure 2-16 Retail processes and BI

These functions and the need for real-time BI will be even more critical as we move forward. Figure 2-17 depicts the types of functions, programs, services, and capabilities that are needed in a real-time environment. The movement to real-time BI is one of the enablers for the real-time enterprise.

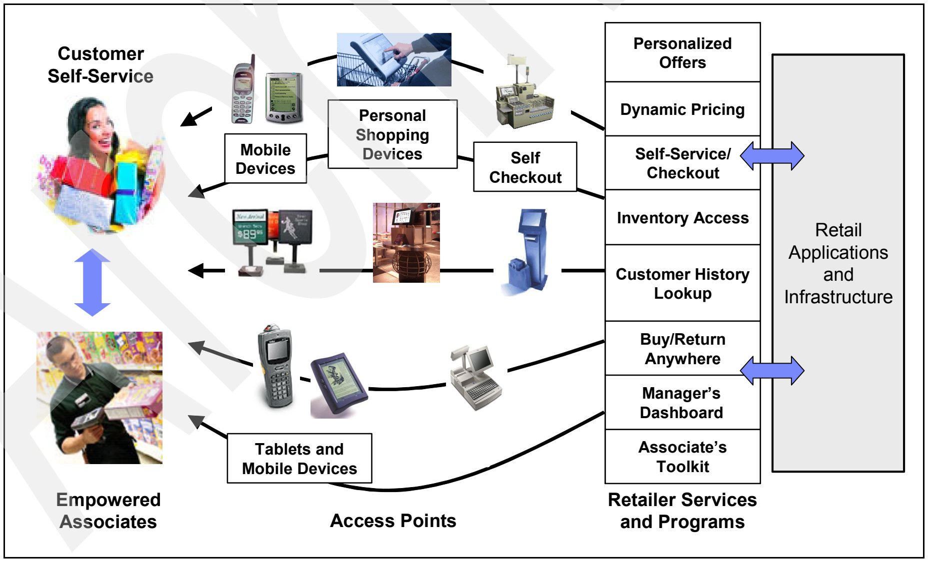


Figure 2-17 Real-time in retail

As examples, personalizing offers and keeping track of consumer behaviors across the various sales access points is typically handled in a central data warehouse. However, you must support real-time delivery of goods, as well as a continuous ingest (or loading) of data to keep up with such important elements as consumer behavior. As technology and server speed advances, dynamic offers to clients via real-time data mining will be a common requirement.

Saving, retrieving, and using the data in the real-time enterprise provides the value. The ability to make key decisions at the right time, and as soon as possible, provides the business advantage. And with the real-time enterprise, you know have the ability to get that *single view of the business* that has for so long been talked about. Such a view, in the retail business, is depicted in Figure 2-18.

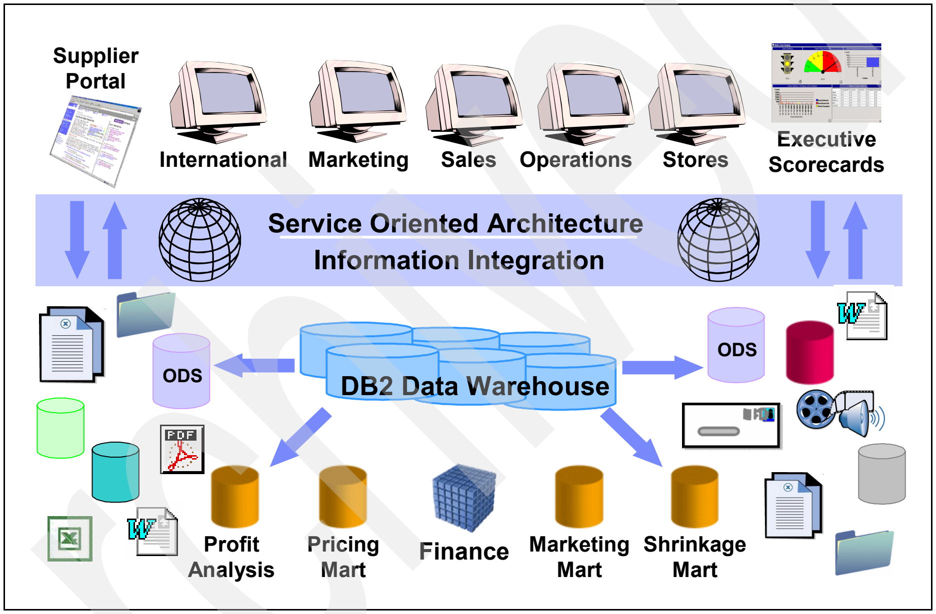


Figure 2-18 Single view of the business

And how are retailers using the information in the real-time enterprise? Here are a few examples:

- Product oriented applications help retailers perform category management, merchandise planning, assortment planning, markdown analysis and promotional planning – including market basket analysis for cross promotion purposes
- Frequent shopper or customer oriented applications help the retailer increase their percent of the *customer wallet share*. This includes activities such as customer segmentation, customer acquisition, customer retention and cross-selling – as well as applications that assist a retailer to execute closed-loop campaigns with customers.

- Systems support decentralized decision making such as space utilization, labor productivity (and fraud prevention), systems deal with issues such as demand profiling and forecasting, inventory replenishment, supplier analysis, logistics/transportation analysis, and warehouse management.
- The BI systems pull information from finance and POS, to enable management to perform timely analysis on daily operations.

IBM has the products, tools, and services to help in implementing a real-time enterprise in the retail industry. We give a brief overview of such a solution in the following sections.

Retail Business Intelligence Solution

The Retail Business Intelligence Solution (RBIS) helps drive the logical approach for delivering data. It provides the foundation and the blueprint for the business enterprise data warehouse. This logical design then drives out a sizing for how to build the data warehouse with a standard building block approach. That solution is depicted in Figure 2-19.

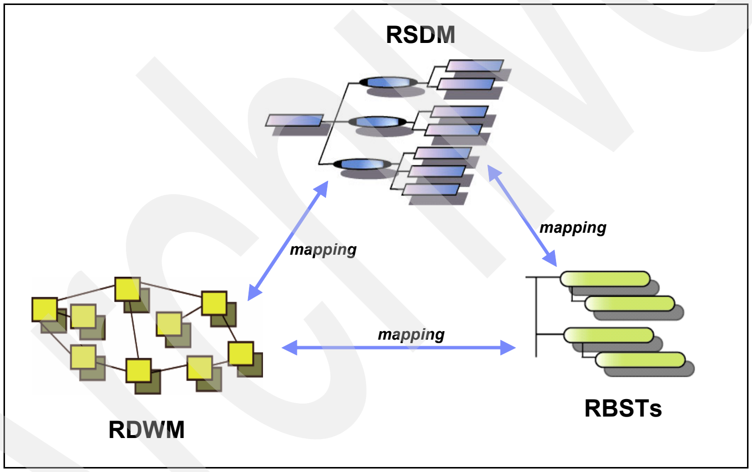


Figure 2-19 RBIS solution

The RBIS solution is comprised of:

- Retail Services Data Model: This is a classification model for defining business meaning across all models and databases. It defines the business meaning and relationships of all the data captured in the RDWM and RBST model.
- Retail Data Warehouse Model (RDWM): This is a Logical Entity Relationship (E/R) model for designing the central data warehouse. It consolidates data from throughout the enterprise into one comprehensive view.

- Retail Business Solution Templates (RBST): These are logical measure/dimension models for defining user information requirements and enabling multidimensional analysis. They are a grouping of measures and dimensions that satisfy a particular business requirement. They organize the data into specific areas for advanced analytics, such as guided analysis.

We mentioned measures and dimensions in the previous list. But what exactly are those? They are terms used in dimensional modeling.

- Measures: These are facts that are used to quantify business performance indicators. They may be made up of other measures (known as sub-measures) in the context of the measure to which they contribute. An example of a measure is n*umber of customers*.
- Dimensions: These consist of criteria or segments by which the measures may be broken down. This includes at least one level of dimension members. An example of a dimension is *time period*.

The IBM solution is unique. There are other solutions in the marketplace, but they consist only of a data warehouse model. The IBM RBIS solution also includes the Model Management Tooling (M1) to fully utilize the power of the structure.

As you can see in Figure 2-20, we are leveraging the Retail model for the data asset layer. The DB2 Data Warehouse Edition (DWE) is used to empower the performance and analytical layer. Then the top layer consists of our best practices and experience with the solution templates. This logical design then enables a sizing for how we build the data warehouse with our standard building block approach.

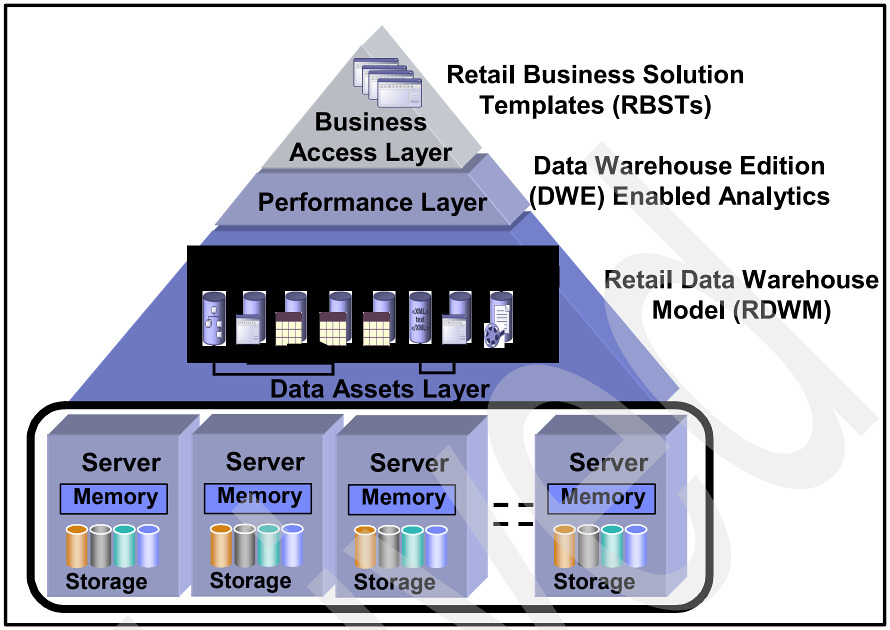


Figure 2-20 Real-time in retail

Now we can give you a different view of the layered IBM architecture, as discussed in Chapter 2. This view shows the same logical architecture but it is overlaid with retail business terms, showing where they appear in the layered architecture. This is depicted in Figure 2-21.

The idea here is to map the delivery of data needs to your applications and organization needs. We call this *right-time data* delivery. As you can see, the lower floors of the architecture offer the most current, and most detailed data to the enterprise. Here is the data you will use to make operational and tactical day to day decisions. As you move up the floors of the layered architecture, the data becomes more summarized and focused for specific business roles or applications. For example, on Floors 3, 4, and 5, you will find users performing more strategic analysis.

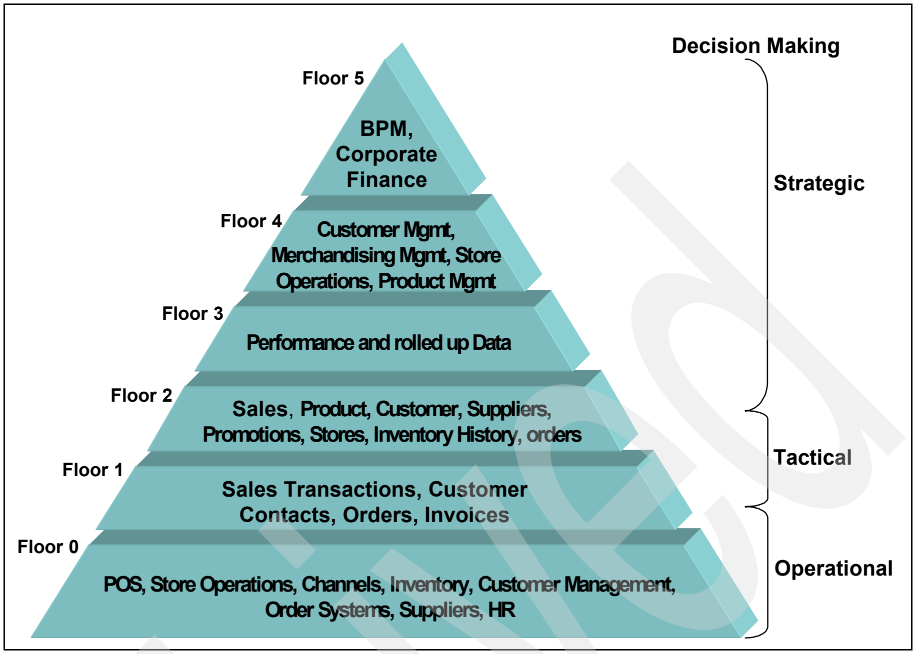


Figure 2-21 Retail layered architecture

The IBM solution will allow you to physically support a design that is tailored to the needs of your business, giving access in real-time to point-of-sale (POS) and inventory data, while at the same time providing summarized views to re-occurring business questions such as:

- What are the sales in this region?
- How do my stores compare in performance?
- Who are my top customers?

This design, when mapped to your enterprise, will help determine the right number and type of building blocks needed to build the data warehouse. The building blocks will be determined by such things as the number of users, complexity of workloads, and how active your warehouse is in terms of data delivery.

IBM has the components that can enable you to implement an on demand real-time enterprise in your retail establishment.

2.5.2 Insurance

The Financial Services industry faces a number of key challenges and business issues. This section will examine those challenges to provide a fundamental understanding of the issues that we help solve with Business Intelligence solutions. In general, this industry has faced a period of industry consolidation due to loosening regulation of bank and insurance companies. While loosening regulation has allowed these companies to expand into other service offerings, other legal and regulatory changes have tightened accounting practices and information privacy restrictions.

For example, there are more-stringent underwriting practices. Underwriting is a core element of the insurance process — and a particularly important concern for risk management efforts — but is typically handled in a largely manual, and therefore less efficient, manner. Insurers will be forced to refine their underwriting practices to increase efficiency and speed, and provide greater accuracy and consistency. Underwriting has gained increased industry attention and is becoming a key part of strategies to reduce operational costs and drive competitive differentiation through enhanced data and segmentation.

Increased use of BI and implementation of a real-time enterprise is required to get the information needed at the time it is needed. This support, and the sharing of enterprise data, must span the insurance value chain, as depicted in Figure 2-22. Implementing this capability is part of becoming a real-time enterprise.

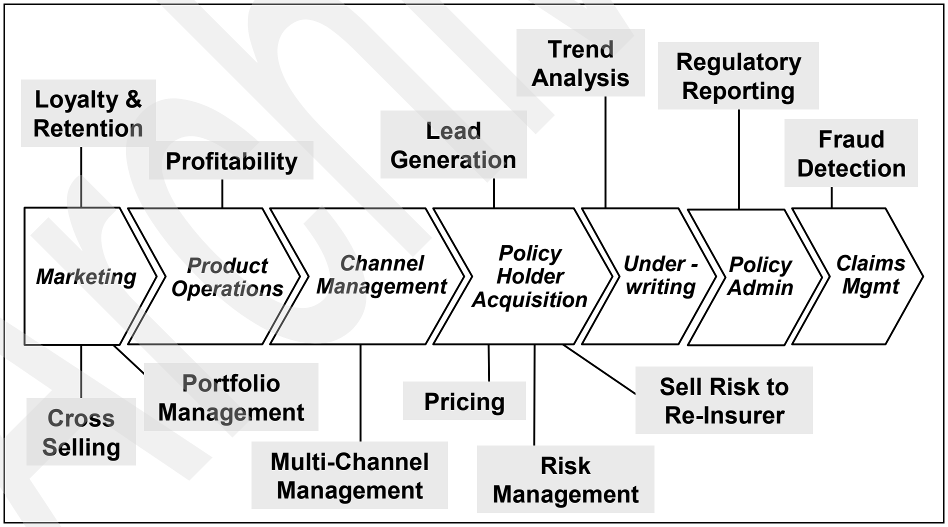


Figure 2-22 Insurance value chain and BI

Policy Submission, Underwriting and Issuance today can be a complex, time consuming and expensive process. Large Insurers may write policies through captive or independent agents who have their own agency management system.

Policy applications are often manually submitted or faxed to the carrier for underwriting approval.

Carriers often have multiple policy systems residing on multiple platforms utilized across multiple departments. Many carriers have implemented content management systems to help manage the volume of paper produced in policy underwriting, particular for commercial policies, but these implementation are typically utilized within a department and not shared across the value chain. Underwriters often utilize separate underwriting and risk management workbenches that require re-keying of data and input of additional data that might not be included on the agent policy application or may require acquisition and re-keying of data from additional sources. Such an environment is depicted in Figure 2-23.

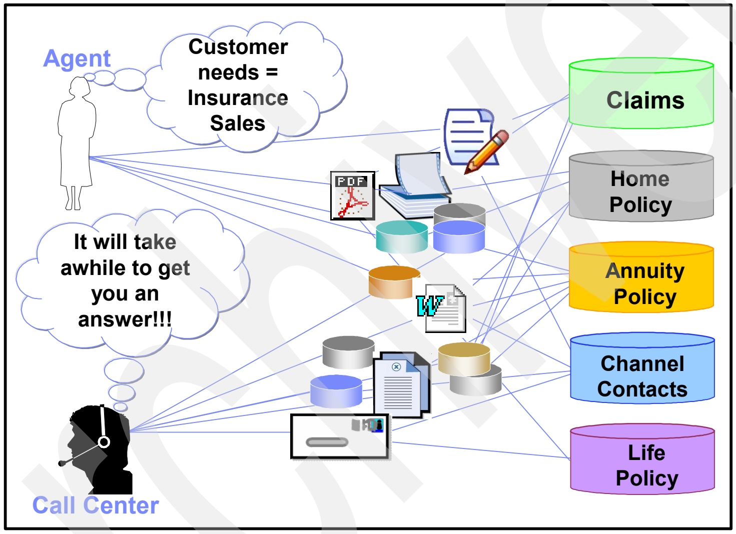


Figure 2-23 Current process

Those insurers that utilize an agent network will find it to be an important data warehousing application in identifying, developing and supporting the top performing agents. Likewise, if the insurance coverage is group insurance provided by an employer, then aggregate losses by all members of the group will typically be summarized and managed at the group level, which will be a fundamental data warehouse application. This is typically the case with health insurance and term life rather than auto, home owner or personal property. However, many of the basic principles are the same with the difference obviously being that group insurance must be administered at the group level — both the payer and the insured must be satisfied with the service in the long run. But if the payer (the employer) is dissatisfied, the consequences are direct and immediate, whereas if the covered (the employee) is dissatisfied, it takes times before the dissatisfaction is brought to bear on the insurer through the employer.

Another important issue in the design of insurance data warehouses is that in property and casualty the lists of covered items can become very large — hundreds of millions of items. The typical approach is to apply database partitioning, multiple data structures (tables) as required and parallel processing in loading and manipulating the data. In all lines of insurance, the product dimension tends to be a lot less stable than in retail. If a group customer is large enough, then the insurance company will customize a product to accommodate its requirements.

This results in a heterogeneous product, which, in turn, consists of a multi-tiered product structure with basic attributes in the first-level table and custom attributes in one or more normalized supporting tables. SQL can handle this quite well, since the coverage key will automatically associate with such insured items as automobile or home or life insurance dimensions.

What is needed is a integrated environment where the data across the value chain is consolidated for consistent access and analysis across the enterprise. Such an environment is depicted with the improved process in Figure 2-24.

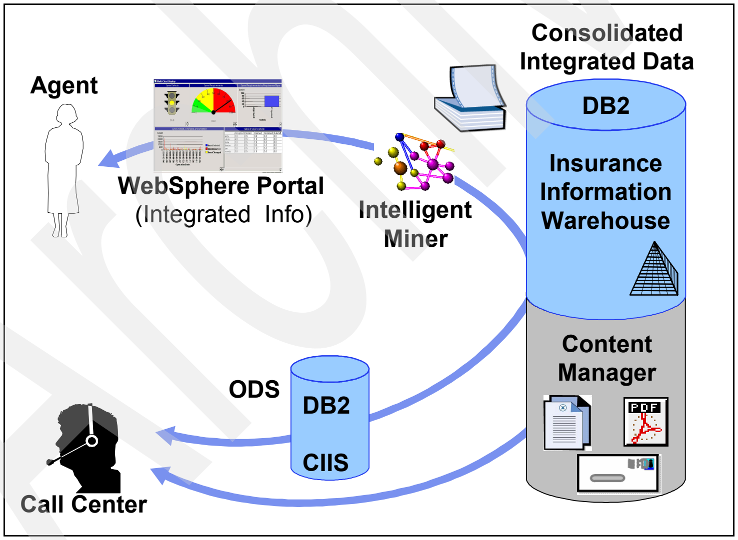


Figure 2-24 Improved process

In conclusion, insurance requirements include basic data warehousing applications, such as knowing who is buying or using what products and when and where are they doing so (selling opportunities). On the product side, the fundamental data cube is what products are profitable (premiums minus losses) by period and customer segment. In the case of group insurance, it makes sense to look at the loss profile by (say) employer group.

When the key dimensions implement consistent representations of customer, product, geography, calendar and other essential entities, then the analytic applications and subject area data marts roll up into an enterprise wide data warehouse. This is the base information required. Access to that data in real-time enables a significant business advantage.

2.5.3 Telecommunications

The telecommunications industry, like many industries, has a great deal of competition, which results in high customer churn. In this industry, customer satisfaction and consequently customer loyalty is essential. A low-latency 360 degree view of the customer can help increase customer satisfaction and consequently reduce customer churn.

The telecommunications industry often develops customer profiles through mining their call data record (CDR) information. To develop these profiles both operational and historic data is used. Integrating these two data sources is necessary to develop an accurate profile. In addition to integrated information the development of accurate profiles requires up-to-date information. Data mining is traditionally performed against data created through batch jobs. These jobs take time to execute so the mining is not done against current data. In a real-time enterprise, the availability of right-time data enables mining applications to use current data and to create profiles identifying the current market trends.

These profiles are often used to predict customer behavior. Integrating these profiles with customer call history enables call center employees to predict what services a particular customer might be interested in. Knowing what customers want, enables the call center to better serve their customers. Knowing what services the customer may want to purchase, enables the call center to target specific customers for specific offers and consequently increase overall sales.

In addition to using right-time data to create the profiles, call centers need up-to-date customer histories to correctly match a customer to a profile. If the customer history is incomplete they may not be correctly matched to the profile that describes them, and consequently an incorrect decision is made. Matching customers to an incorrect profile is especially likely to occur in the case of a new customer.

Since there are times when customers are dissatisfied with services, it is sometimes necessary to offer pricing deals or services in order to keep a customer. Combining current customer information with mined profiles allows the call center to determine the value of the customer they are currently dealing with.

Knowing the relative value of the customer to the company provides a framework of offers and services that the call center can extend to keep the customer.

The fraudulent use of communications is a common occurrence in the telecommunications industry. Fraudulent calls reduce revenue and use available resources. One technique for detecting fraud is to create profiles for each customer and then compare the profiles with the CDR data. These profiles include data about who the customer calls, when they call, and how long their calls last. These profiles, like the general profiles mentioned early, are most useful if they were created from mining up-to-date data. The early detection of telecommunications fraud enables a company to reduce the impact on revenue and resources. Comparing right-time operational data with up-to-date profiles insures that the fraud is detected as soon as the behavior captured in the CDRs does not match the expected behavior in the profile. The availability of up-to-date customer profiles and up-to-date CDR information reduces the likelihood that a customer's service will be interrupted because it is falsely believed that they are the victim of fraud. Falsely identifying fraudulent use of a service will negatively impact customer satisfaction.

The nature of the telecommunications industry requires cooperation between carriers in order to deliver phone services to their customers when they are away from their home area. To accomplish this cooperation, telecommunications companies share information. Even with this sharing of information, there are still cases where revenue is missed because a transaction is not identified. This issue can be addressed by comparing transactions billed with overall transactions. Integrating the billing and CDR system enables telecommunication companies to determine whether a call has been billed. This information can also be used to determine the correct charges in cases where the telecommunication company is being billed for the use of another companies network. Due to the volume of these transactions, small mistakes in billing can quickly add up to large revenue loss. Timely identification of billing mistakes requires that both right-time transaction and billing information be available.

Telecommunication companies utilize their CDR data to understand their business. CDR data is used to identify market initiatives, detect operational failures and provide service offerings. The ability to mine right-time data produces accurate profiles which enable executives to understand their customers and identify marketing opportunities. Up-to-date profiles allow companies to target specific market segments and create specific marketing plans.

Up-to-date profile information also allows executives to analyze customer churn. The ability to mine right-time operational data based on CDRs can provide accurate and up-to-date customer information which can be used to identify key indicators of customer churn. Mining right-time data can also identify customer calling habits and identify the most profitable customers. Customer models can be created from this mined data. These models can provide c-level executives with a complete profile of their customer and enable them to make marketing, pricing policy and service offering decisions. These models can also be used to better understand customer churn. Understanding customer churn enables companies to target marketing to specific customers. Understanding churn also enables a company to predict churn and target services to retain valuable customers.

The availability of right-time data can help telecommunication companies increase customer satisfaction. The integration of various data sources provides a complete view of the customer, which enables executives to make both tactical and strategic business decisions. The combination right-time data and systems and application integration enables telecommunication companies to better market themselves and increase customer loyalty. These types of companies are real-time enterprises.

2.5.4 Travel

The travel industry books various activities for their customers. These activities range from hotel accommodations to cruise bookings. Their is a many to many relationship between travel agencies and travel resources. A single travel resource supplier works with several travel agencies and a single travel agency works with several resource suppliers. In this day and age, booking information is exchanged electronically. This means that many different systems must interact with each other in order to book a resource, requiring both systems and data integration. The integration of heterogeneous systems enables travel companies to get the data about available services and resources from other travel companies, and offer them to their customers.

In order to successfully provide services to their customers travel companies need up-to-date information. Since many resources in this industry are booked from many different locations, up-to-date information is needed to confirm the true availability of the resource. Right-time information is required to insure that the needed resource, such as a room, flight, or cruise, is not already booked. Double booking a resource will result in poor customer satisfaction and could impact the future sales.

In recent years it has become common to book travel resources through the internet. The appearance of internet portals and internet travel companies has added many new customer channels. Travel providers can receive resource requests from several internet sources in addition to their own Web site. Also since the travel industry is so information oriented and electronic, established travel booking companies are now competing with companies who provide services only through the Internet. This increase in service providers requires that the travel provider data systems be integrated across many different systems to verify resource availability. Since a single customer trip can require several different resources, a competitive travel company must be able to search through available resources and match their resource offerings to the customer's requirements.

The availability and popularity of internet booking has greatly increased the competition for customers in this industry. Consequently satisfying and maintaining customers is critical. A 360 degree view of the customer is needed in order to provide the best possible customer service. However, getting such a view is a challenge. The reason for this is that the customer experience includes interaction with several resources from several different companies. For example, a single customer trip requires interaction with a transportation company and a hotel company. In addition, the customer may also interact with a car rental company and several tourist oriented service companies. Getting a 360 degree view of customers in this environment requires the integration of data from a number of systems. One way to do that is to mine (data mining) the data and create individual customer profiles.

The travel industry like many industries depends on market trends. Certain activities and destinations become fashionable each year, and others fall out of favor. In addition, there is a seasonal aspect to travel. It is a constantly changing industry. Due to this nature, right-time information is a necessity. It can provide more accurate profiles, and enable companies to predict market trends and provide service offerings that are in most demand.

The nature of the travel industry requires that both travel booking companies and travel resource providing companies cooperate. In addition, travel resource providing companies must also cooperate amongst themselves. An easy example comes from the airline industry. There are times where it is necessary for a carrier in one country to cooperate with a carrier in another country in order to provide the customer with service to their destination. To accomplish this it is necessary to share information. Even with this sharing of information, there are still cases where revenue is missed because one air carrier collects the fees and must pay another air carrier their share of the overall fee. This type of revenue leakage can be addressed by comparing transactions billed with overall transactions. Integrating the billing and operational data system enables travel companies to determine whether a travel resource has been billed. This information can also be used to determine the correct charges in cases where their is a dispute between the company being billed for the service and the company that provided the service. Since this type of resource utilization is common in this industry, small mistakes in billing can quickly add up to large revenue loss. Timely identification of billing mistakes requires that both right-time operational information and billing information be available.

When data is shared across several companies and, and perhaps several countries, a number of heterogeneous systems are involved. So it is critical that the integration capability be robust to eliminate the possibility of revenue leakage, and enable travel companies to better plan and provide the required future services and products. This complete view of the customer and robust heterogeneous integration can enable companies to provide the best service possible for customers in the travel industry.

Summary

This chapter has described some of the characteristics of a real-time enterprise. It is about getting data into the enterprise data warehousing environment as fast as is needed and financially justified. That makes right-time data available for business intelligence, that enables improved decision-making and business management. That closes the loop between defining business objectives and measurements and their attainment.

3

Architectural considerations

The real-time enterprise is a significant initiative in business today. It is no longer a question of whether you need to move in that direction, but one of how fast you can progress towards creating the infrastructure and capabilities to support such an environment.

In this chapter we describe some key considerations, and some of the key component functionality required, as you begin developing an architecture to support your real-time enterprise. We also discuss the demands for the various infrastructure technologies to support the real-time enterprise that are available today. And lastly we look at some future trends and directions, and the next generation of infrastructure components.

3.1 Introduction

Implementing a real-time enterprise environment is no simple task. It will take time, expense, and energy. But, having such an environment will pay huge dividends in terms of business viability and competitive business advantage. And one of the first things you need as you start your journey, is a good roadmap. In terms of this initiative, that roadmap is actually an architecture.

The objective of this chapter is not to present you with such an architecture, but rather discuss the drivers, requirements, and a framework for one. You will then need develop your architecture based on your specific requirements.

From our perspective, there are three major drivers to be considered when developing a real-time enterprise architecture. They are:

1. Data silos
2. Isolated decision making
3. Latency in the decision making process

The following is a brief discussion of those drivers.

Data silos

Much of the data available in an enterprise today is housed in what we call *data silos*. These are sources of data that have been collected and managed by specific business functional areas. They are typically not integrated with other data in the enterprise, and, in fact, may not even be shared with them. And the data in these various silos is typically at differing levels of currency, and may well even have differing data definitions (metadata) for the same business entities.

Having these data silos at differing levels of currency, and un-integrated, are major reasons why there is inconsistency in reporting and data analysis in businesses today.

Data from these diverse sources typically takes a long time to be consolidated into a consistent, unified global view that allows the business analyst to discover potential problems, take corrective action, identify new business opportunities, or detect anomalies. In many cases, the massive amounts of data from brick-and-mortar stores, Web sites, call centers, inventory, and CRM systems are often analyzed independently from a business unit perspective, rather than integrated to realize an enterprise-wide view.

This is typically because the systems have been developed independently. As such, they likely do not communicate with each other. And the data from these systems may be represented in different formats, stored in different databases and captured in diverse applications.

There is a significant effort required to cleanse, reconcile, and integrate data from the various data silos around the enterprise. However, to move to a real-time enterprise environment, there is a requirement to do so.

Isolated decision making

In a traditional business environment, business planning or financial modeling is typically an offline activity. Statisticians would gather and analyze huge volumes of data from the enterprise data sources and generate a stack of reports representing the results of their analyses. Then business analysts would interpret those analyses and develop a business model and plan. The management executives and officers then develop the budget based on data analysis and their business intuition. As you can see, the activities are exclusively disparate and disconnected. With any luck, the offline reporting and offline analysis might provide companies with the expected value. Without luck, there is a good chance that they will not be able to provide accurate and consistent information for adequate business intelligence purposes.

Latency in the decision process

Most businesses would reap tremendous benefit if they could reduce the latency between business events, information availability and access, and the formulation of responses. For example, the slow sale of one product in one region might be trivial and be overlooked. However, if it repetitively occurs in many stores, the impact could be significant and require fast investigation. Unfortunately, the business intelligence information needed to detect market shift and customer attrition is too often not available to the decision makers in time to initiate a proactive response. As a result, companies are more often reactive in an attempt to minimize the impact of events rather than being able to avoid them.

3.1.1 The impact of data warehousing

Data warehousing has, over time, become the defacto primary information environment for businesses. As an architected environment, it is flexible to enable change, while at the same time able to deliver the capabilities needed to support analytic applications, queries, and all forms of data access and analysis. With the increased need of making data available faster, the data warehouse environment now includes the operational data store (ODS). The challenge is to architect such that the contents of the ODS and data warehouse can be integrated to enable the required support for a real-time enterprise environment.

Operational and informational

The first and most fundamental paradigm of data warehousing is so obvious that it is often overlooked. Data warehousing has always made the assumption that the world of information processing can be split into two parts, operational and informational.

These two clearly related areas, essentially areas of business activity, are assumed to operate largely independently of one another. This enables - and indeed drives - the existence of two separate spheres of IT implementation:

- Operational systems responsible for running the business
- Data warehousing to provide information to enable business decision making.

This split was clearly advantageous for IT because it protected the operational systems from the performance and security impact of the large and long running ad hoc queries that are typical of the data warehousing environment. It also suited a business model in which different departments and organizations operated as distinct and independent business units.

In the early 1990s, the data warehouse environment itself provided two conceptual sets of function:

- The business function supporting data analysis and decision-making
- An underlying and largely hidden function of resolving differences between the detailed world of operational systems and the wide variety of informational needs of the business

These two functions were supported by derived and reconciled data respectively, and came to be known as data marts and the enterprise (or business) data warehouse. The need for a reconciled data layer is also based on the assumption, in this case, that the data in the operational environment is too complex, disjointed, or un-integrated to be easily used by business users.

This conceptual division of the world of data into three areas became the basis for the physical implementation of the three-layered architecture, consisting of operational systems, an enterprise data warehouse (EDW), and data marts, with batch-based extract, transform, and load (ETL) processes linking the layers.

It has become increasingly obvious in recent years that this architecture cannot meet all business needs. It is particularly stressed by the requirement to evolve to a real-time environment.

Closed-loop processing

Closed-loop processes are those where the results of an action are fed back to the operational process activities to impact and enable change to the process. These processes and their IT support must be flexible for change. But by closing the loop, there can be continual process improvement, in terms of efficiency and effectiveness.

Data volumes

We all know that data volumes are growing at an ever increasing rate. And, the data and content mix are ever changing. For example, companies are having to deal with more and more unstructured data. And the volumes of unstructured data are often so large that making and maintaining copies can be prohibitively expensive. Differing technical requirements for storage or processing make it difficult to create a combined store of structured and unstructured data. In addition, legal and privacy constraints are often onerous for such content.

Total cost of ownership

The total cost of ownership (TCO), including servers, disk, and personnel costs, can be reduced by driving down the number of copies of data, and the number of redundant data sources, in an enterprise. There is a significant movement in the marketplace towards data consolidation. And, with the vast improvements in servers, databases, and network speeds, there is seldom a need to distribute data for performance reasons.

Application development performance can also be enhanced because changes to applications can be accommodated more quickly by altering views in the database than by modifying ETL and physical data models, and then reloading the data marts. This can be particularly important during the early stages of the application life cycle, when needs are being dynamically discovered. In rapidly evolving businesses, this benefit can pervade the entire life cycle of the application. Recently, the industry has focused on pervasive business processes and the need to monitor events and enable proactive corrections through business performance management. This raises even more questions about the traditional data warehousing approach, and prompts thought about the potential requirement for change.

3.1.2 Information management challenges

Integration has become a key requirement for virtually every company, particularly those dealing with mergers and acquisitions, and the explosive growth in data volumes. This is exacerbated by all the data within companies that exists in independent data silos and disparate applications across the business functional areas.

According to IBM research, more than 60 percent of CEOs state that they need to do a better job capturing and understanding information rapidly in order to make swift business decisions, a key to the on demand business. In this context, it is not surprising that as much as 40 percent of IT budgets is going to integration projects.

There is also confusion in the market with many vendors using different terms for very similar solutions. As a result, organizations use a great deal of resource trying to understand these different vendors offerings and value propositions. One approach to help here is to ask the vendors for their *best practices* documentation. If they have none, perhaps you should proceed with care.

Vendors should also have an architecture for their solution. Throughout this chapter we provide examples of some of the IBM best practices and discuss some of the component architectures for implementing a real-time enterprise. We also make suggestions regarding specifics areas that should be evaluated very carefully when choosing components for your real-time enterprise solution.

A mayor challenges for all organizations is that they have many disparate information sources located within and without their enterprise. They have such sources as file based data, content repositories, relational databases, and application repositories. And they have them on many different platforms and technologies that have different semantics, data formats, and structures. Having these disparate information sources represents a significant challenge to the task of enterprise information integration.

Organizations not only have a need to access these disparate data sources, but also a need to make them work together, or integrate them. Figure 3-1 illustrates the requirement that different business systems have for accessing their own data as well as that distributed across many other business functional areas. Mergers and acquisitions adds to this complexity.

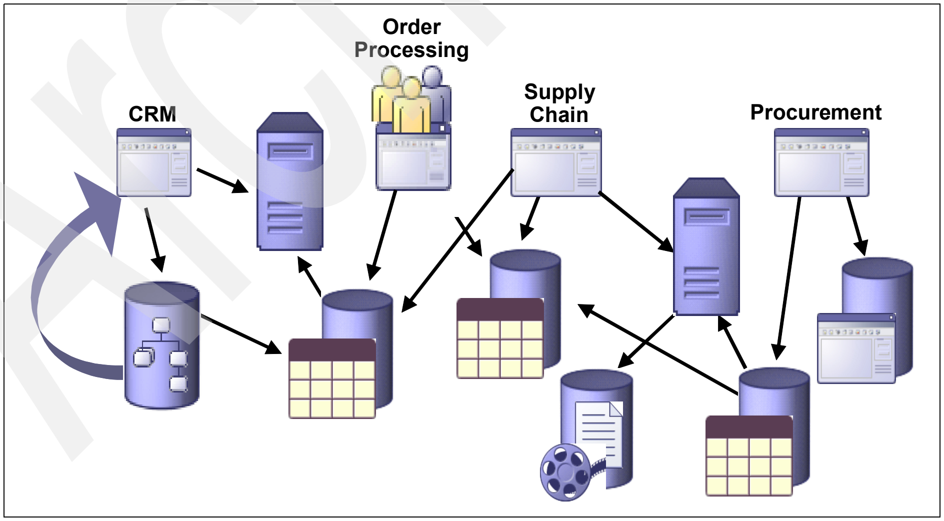


Figure 3-1 Distributed islands of data

The IT organization

IT organizations today are aware of the value propositions provided by information management technologies. The focus of most discussions has shifted beyond information management itself to the larger issues surrounding the development of an overall architecture for implementing a real-time enterprise.

Data warehousing technologies remain a vital foundation for any strategic approach to leveraging all of an organizations information assets. It establishes consistent, predictable levels of data quality, consistency, and stability.

However, the scope of business intelligence architectures has expanded beyond traditional data warehousing disciplines such as data extract, transform and load (ETL) and reporting. Today, business intelligence requires information *on demand*. This may include actionable business insights from combining functions such as data mining and multidimensional data analysis, with advanced statistical and analytical functions in a low latency real-time, integrated environment. OLAP and data mining, in particular, have brought with them specialized tools, APIs, and data structures. And also purpose-built engines, spanning all levels of the system infrastructure, from front-end client tools, to middle-tier server cache, to back-end data warehouse.

The multi-tier, enterprise-wide nature of BI tools and infrastructure demands a high-level architectural model, a framework for BI. A BI framework provides an overall conceptual model for understanding, planning and managing this complex topology. The BI framework should express the strategy, ideally in a way that differentiates the particular approach from others. It should also be capable of enlightening customers regarding what to expect in terms of future direction, technology and interfaces. And, it should guide system architects and integrators in selecting, designing and deploying BI tools and applications.

Data governance and documentation

Data governance is required in any organization to ensure consistent data quality and meaning. Having good standards for formatting, key matching, and distributed maintenance is still a big issue in many organizations.

As systems evolve and data changes, perhaps simply moving to a new operating environment will result in changes to the meaning of the data. Therefore, good documentation should follow and be kept updated.

3.1.3 Creating an information infrastructure

In the following sections we illustrate solutions that will provide resolutions to the information management challenges. Here we look at how to enable a business application that has the requirement to access consistent information with a holistic view of all data inside and outside the boundaries of the enterprise.

Such a complete information strategy must be built on an extensible foundation with rich capabilities supporting innovative business applications. Such a strategy implies an information infrastructure that can simplistically be thought of as having two parts:

- Information assets
- Innovative information services

The information assets can include information managed in databases, file systems, data warehouses, and content repositories. Robust asset management includes critical capabilities such as data sharing, performance, scalability, availability, and security. Additionally, and increasingly more important today, are a set of information services to support applications that span and leverage the enterprise information assets. These services include search and retrieval, integration, and analysis and content control, along with industry-specific extensions.

Enterprise Application Integration (EAI), Enterprise Information Integration (EII), and extract, transform, and load (ETL) solutions are based on different architectures and are typically used by different organizations in an enterprise. Do they have the same understanding of what the data means? Or do they understand what the key performance indicators are after data has been delivered through either EAI or ETL? These things we need to ensure.

Layered data architecture

To enable the integrated view of data assets, and consistent and easy access and use, we suggest a layered information architecture model.

Most organizations have many different data repositories and databases, often on many different operating platforms. There are many different people and functions accessing the data on both an ad-hoc and regular bases. They may be transforming or enriching the data to be able to fulfill their business needs. And typically this can result in the meaning of the data getting changed or diluted. All this can make the integration requirements very challenging.

A common business and application access layer, and data model, is important to enable a consistent view of any organizations data. Performance is also a key criteria. There is a requirement to deliver a well performing access layer to the business layer without putting unnecessary and disrupting workloads on the operational systems. So, all business applications should access the data layer through the performance layer, either with EAI or EII. In most situations, the business application will not have the need to go all the way to the operational layers.

Until today many business intelligence environments have been seen as standalone systems. They have had specific dependencies on various operational systems being capable of delivering data to the data warehouse in a timely manor. And, they have typical been used for business areas such as point-in-time reporting and detailed ad-hoc analytics. They are also batch oriented systems which are updated on a monthly, weekly or daily basis. The data warehouse has so far had a highly specialized usage, which is supported by the creation of data marts or cubes to satisfy specific analytical business needs.

New business demands require the business intelligence environment to be capable of supporting the more diverse workload of the real-time enterprise. The data warehouse should be capable of supporting a real-time analytics environment as well as providing the traditional historic representation of information. At the same time, it also needs operate in an integrated environment. Thus the need for a layered data architecture. We represent such an architecture in Figure 3-2.

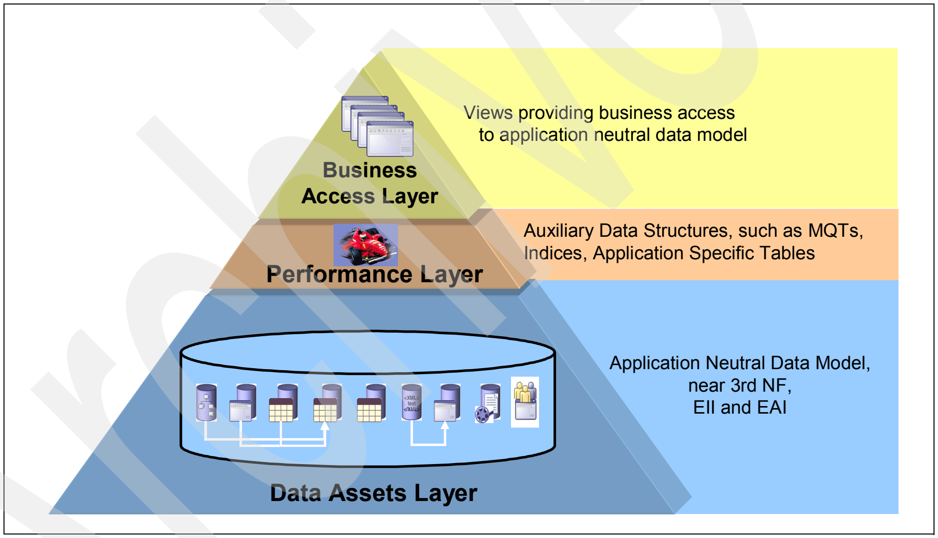


Figure 3-2 The layered data architecture

3.1.4 The layered data warehouse architecture

The same need can be applied to data warehousing. A layered data warehouse architecture can provide a logical view of how an enterprise data repository is built, and how it enables analytics and other key business applications.

By layering the data warehouse it enables layering in functionality without requiring the separation of distinct subject areas. It is then based on a scalable infrastructure that allows for dynamic redistribution of resources and functionality, as well as capacity on demand. They key is to leverage an enterprise set of data, and build in a performance layer to enable analytics. By leveraging the deep BI functionality of DB2, OLAP performance can be improved and data mining operationalized in the performance layer.

In Figure 3-3 we illustrate an architecture, or information pyramid, that many organizations already have today, at least to some degree. They have different layers and usages of systems, ranging from operational, through ad hoc analytics, to high level dashboards providing high level summarized information for business decision makers.

The various layers in the information pyramid are typically oriented to different user groups in an organization. This is illustrated, in Figure 3-3, by the various groups or purposes of the data warehouse environment.

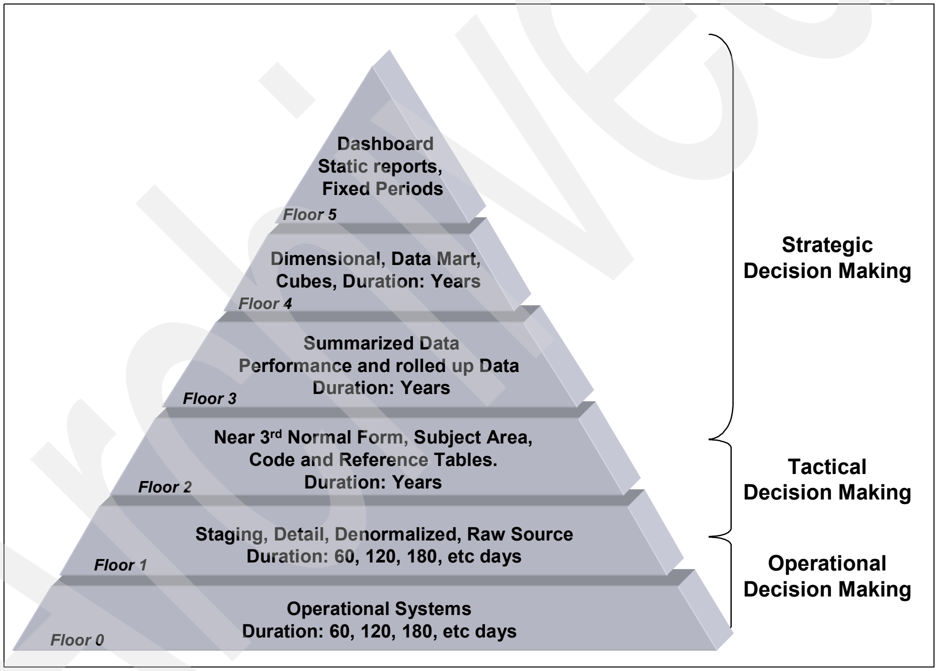


Figure 3-3 The IBM layered data warehouse

We refer to the various layers of the architecture as *floors*. Each has a different use, form, currency level, and duration of data retention, to support the various requirements of the enterprise. We now give a brief description of each of the floors.

- **Floor 0:** Here we find the operational systems, such as point-of-sales, store operations, channels, inventory, customer management, order systems, suppliers, HR, and billing systems. The duration of the data is measured in some increment of days, such as 60, 120, and 180.

These systems deliver data to the upper layers of the data warehouse architecture model. The duration of the data life-cycle at the operational level can be short. The operational data layer itself does not often need to keep data for periods longer than the current month, quarter or year. The life-cycle of the data at this layer is dependant on capabilities that the higher layers provide. If, for example, any operational end-of-year reporting is designed to be done directly at the operational layer on floor 0, the data must be kept for such purposes. The data at this layer are normally created at a very detailed level.

- **Floor 1:** This layer is what we call the operational data store (ODS). Examples of the data kept at this layer are sales transactions, customer contacts, orders, and full invoices. The duration of the data life-cycle on this layer is generally very short. Few business applications accessing data at this floor have a need for long term storage of the very detailed transactional data.
- **Floor 2:** Here we typically talk about a normalized enterprise data warehouse structure where we have reference data that enriches the transactional data to enable use by analytic business applications. Data at this floor may be stored for many years. Particularly after some of the recent regulatory compliance specifications, the enterprise data warehouse has become the primary vehicle to support these requirements.
- **Floor 3:** This floor can be thought of as the performance layer. We have a requirement to provide satisfactory performance of the data access layer to the business application layer. We will at this layer monitor and maintain the performance of the data access layer. As business applications develop and change over time, it is important to have a good solid strategy for how this layer is architected. It is very easy to start moving data out into many different summarized subject data marts, but this can be costly to develop and maintain. A good alternative to isolation of stand-alone summarized data marts is to keep the summarized data as close to the data warehouse environment as possible. This approach can make it easier to provide the above mentioned optimized access to the data at the lower layers.

  The data typically kept in the enterprise data warehouse is detailed historic information about such topics as sales, product, customer, suppliers, promotions, stores, and inventory. All these data are typically in structured data formats, with such things as timestamps, characters, and numbers.
- **Floor 4:** At this floor we provide access to data either by delivering specific business oriented data marts, or through a set of integration technologies. A challenge for this floor is maintaining large amounts of redundant data. It is important to deliver a consistent view of the data. So the primary purpose of this layer is to provide reporting systems and to enable ad-hoc analytics.

Examples of systems being supported at this layer are customer management, merchandising management, store operations, and product management.

- **Floor 5:** At the top layer we see corporate dashboards supporting top level management. The data for the dashboards can be accessed directly at any of the lower data layers. At this floor we also typically see static reports that support many levels of management.

With these differing layers of data, we can more easily define and support the functionality provided by the data environment. To do so, typically requires access to data across the multiple layers, and movement, and perhaps transformation, of data between the layers. We now look at the approaches for such data access and movement.

3.2 Approaches for data access, transformation, and movement

An important part of the real-time enterprise architecture is how we enable systems to either access data anywhere or how we move the data around the enterprise. The goal of information integration is to deliver the right information, in the desired form, when it is needed. By doing this, the applications are isolated from any underlying information complexity and access challenges. In this section we look at the different architectures available to support this.

3.2.1 Extract, Transform and Load - ETL

ETL tools have been the workhorse of data access and integration for years. They were created to extract data from a variety of sources, transform it as required by the underlying data model, and then load it into the data warehouse, typically in a batch mode during non-operational hours. The data volumes involved are generally large, the load cycles long, and information in the data warehouse typically a day to a week old. For consolidating and staging data across operational systems for tactical tracking and decision making, operational data stores were created. They then were used as a source to populate the data warehouse.

Now with the advent of right-time business intelligence, the ODS is used in conjunction with the data warehouse to enable the real-time enterprise. And, designed to process very large volumes of data, ETL provides a suitable platform for improved productivity by reuse of objects and transformations, strict methodology, and better metadata support, including impact analysis. Traditional ETL tools have matured over the years and are now enhancing their support to include event-driven data and Web services based data sources. Figure 3-4 shows an overview of the ETL process architecture, supporting a number of different heterogeneous data sources.

There is also an extended version of the ETL architecture called ELT or ELTL, depicted on the right side of Figure 3-4. Here the data is extracted from the data sources and loaded directly into and ODS. This can enable faster data load times. Then, any required transformation is performed. It too can be fast because we can take advantage of set processing rather that transforming and loading a record at a time. This approach can also utilize the capabilities of a highly parallel database engine.

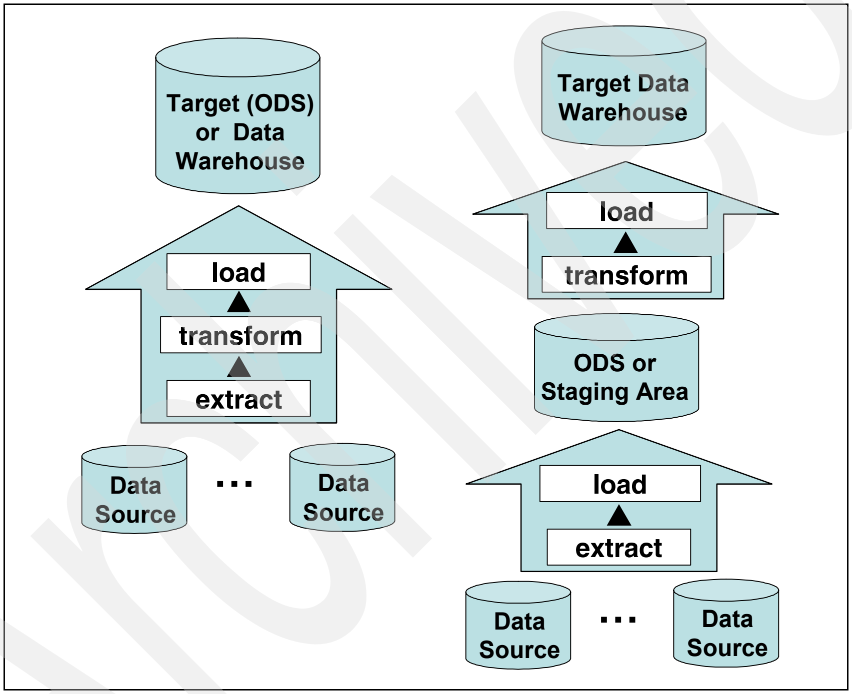


Figure 3-4 Data access and movement

3.2.2 Enterprise Information Integration

Enterprise Information Integration (EII) enables access to information, in the form desired, when it is needed, while isolating applications from the underlying information complexity. Delivering this capability requires an integration platform that supports the complete information delivery process by connecting to any data content, enhancing the data for accuracy and consistency, and delivering timely data.

Information integration is used in critical business initiatives such as business intelligence, risk and compliance, corporate portals, infrastructure rationalization, master data management (MDM), and business transformation. While organizations use EII to solve specific business problems, they are increasingly recognizing the significant value of combining these capabilities in powerful and innovative ways to leverage synergies and ultimately create strategic advantage.

Likewise, as organizations expand deployment and usage of information integration products across the enterprise, they are asking for tools and capabilities for streamlining design and development. In other words, they are asking about capabilities regarding existing information assets (leveraging what they have) to simplify design and development.

Data continues to explode in volume, velocity and variety. As an example, a large volume of the data created today is unstructured. EII provides an excellent solution to retrieve both structured and unstructured data.

Figure 3-5 shows an overview of the architecture of an EII environment, where a data virtualization layer enables access to heterogeneous data sources.

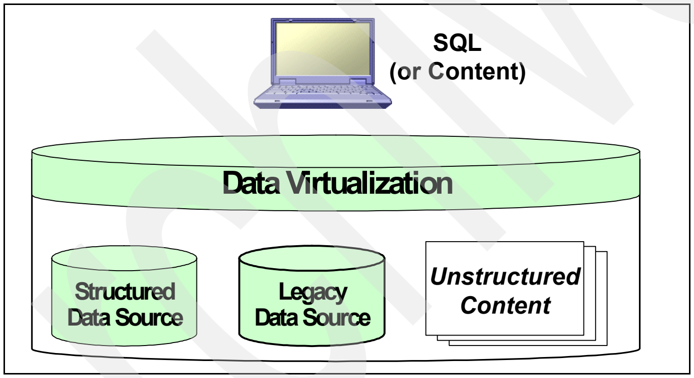


Figure 3-5 Overview of EII

An EII engine should be designed to meet a diverse range of data integration requirements, and should provide a range of capabilities such as:

- Enterprise search
- Data federation
- Data transformation
- Data placement (caching and replication)
- Data event publishing

It should also add value to most integration projects, such as:

- Building a corporate intranet

- Augmenting a data warehouse with real-time data
- Building a unified view of customers or products
- Managing data consistency, distribution, or synchronization across applications

Scalability and usability

An EII engine can scale to support potentially thousands of users across potentially millions of documents and data records. It can also integrate into already existing enterprise applications through open standards. With these capabilities, an EII engine can improve performance, potentially by an order of magnitude for key usage scenarios. And, it can also extend the application development efforts by providing a data virtualization layer.

When to use EII

Information integration can be used for the following types of applications:

- **Creating a single view of a customer or other business entity:** Many business entities require data from different data sources. And many areas of this single view require access to in-place data. That is, access to data where it currently resides - such as the current inventory. To support this capability efficiently, requires federated data access.
- **Enterprise data inventory and management:** One very important data integration capability is to maintain an inventory of data elements, with respective metadata, to guide future integration efforts. By providing direct access to the sources, EII is well positioned to enable this capability.
- **Real-time reporting and management dashboards:** Monitoring and operational reporting relies on up-to-date information to provide a comprehensive view of the data. EII can provide an integrated data layer to support such applications.
- **Data exploration:** When in the development phases of the semantic definition process, EII provides a quick and efficient way to test hypotheses and resolve conflicts with a definition of integrated data views. By performing data exploration using EII, many pieces of the semantic layer will be developed and can be reused later in combination with ETL and/or EAI development.

Information integration simplifies the creation of all these applications by enabling the information to be accessed and managed as though it came from a single data source.

EII has advantages and is best used when source data:

- Volatility is high

- Selectivity is granular
- Connectivity is reliable
- Service levels are compatible
- Transformations are minimal and can be expressed as SQL

3.2.3 Enterprise Application Integration

Batch ETL solutions are minimally capable of meeting the real-time integration requirements of the real-time enterprise. Information that is a day old, for example, may not satisfy the requirements of the real-time enterprise. While the ETL tools continue to serve a valuable function in organizations, they require added capabilities to support the requirements of integration.

Enterprise Application Integration (EAI) solutions came along and solved some of the data latency problems by synchronizing changes across systems in real-time. EAI is focused on the integration of data among a collection of applications or systems. As data is changed in one system, the change is propagated to other systems of interest, typically via asynchronous messaging. Application integration, though required by business functions, is primarily the domain of the IT organization. It is the responsibility of the EAI systems to keep these various systems in the organization synchronized with each other.

The issue with some application integration platform products is scalability and transformation power. EAI less adequately addresses the need to aggregate and consolidate data and information across the enterprise. EAI can effectively move data among systems in real time, but does not define an aggregated view of the data objects or business entities.

Figure 3-6 depicts an architecture overview of an EAI environment, where there is a service message broker, or service bus, that is interpreting messages and routing them to the appropriate applications connected to the EAI engine.

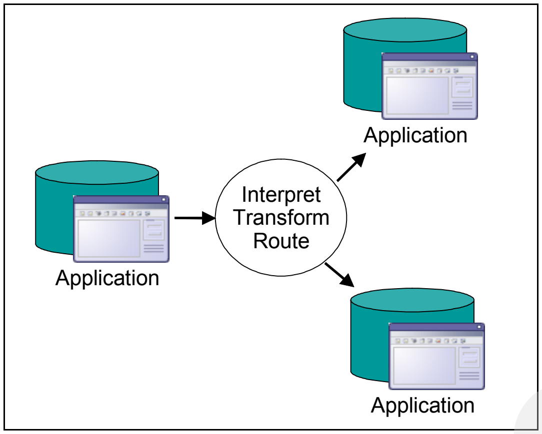


Figure 3-6 Overview for EAI

An EAI approach is widely used today when assembling solutions that must compensate for the proprietary or incompatible APIs exposed by existing packaged applications or custom-built systems. EAI solutions typically consist of a number of proprietary elements that cannot usually be mixed between EAI products, including:

- Hubs to simplify the number of connections
- Adapters to transform interfaces and data formats
- Process orchestration
- Tight coupling to messaging technologies

The adapters often have to use existing system APIs that do not necessarily provide an ideal route into existing applications, particularly when the needs of the real-time enterprise are considered.

Although there are some EAI and messaging *standards*, these are not universally adopted. Consequently, the proprietary nature of EAI can result in high implementation and product acquisition costs, and while it provides some resilience to change by offering new adapters, the overall solution typically remains tightly coupled.

EAI and Web services

As well as using a distributed computing approach as already outlined, Web services can also be used in an EAI style of integration. The need to use EAI to compensate for proprietary APIs diminishes as:

- Package and platform vendors provide standardized Web service APIs in their products, reducing the need for EAI adapters.
- BPEL is adopted for standardized process orchestration.

A hub architecture may still remain as a focal point for coordination and orchestration, particularly where there is a requirement to mix Web services and other non-Web service resources, even though other patterns may become popular. Additionally, Web services do not overcome the semantic differences in information that must still be compensated for, even if all the messages are in XML.

3.2.4 Data replication

Data replication is for many information system implementations an old and proven architecture. It has for many systems been the only way to achieve real-time data movement between non-related or application connected systems.

The replication architectures of today are typically closely related to the chosen database technology. And, traditionally these different technologies have implemented data replication differently.

Many database vendors have had very little focus on further enhancing and developing the data replication functionality within their database. Other vendors have seen the replication functionality as the core technology part of the information integration requirements that have more and more become requirements for all IT technologies vendors to support.

The most common data replication architectures are message-based replication or connection-based replication handled by change detection. With these architectures the implementation is primarily a trigger based capture of changed data or alternatively a transaction log based data capture.

For high data volume replication demands, the log based change data capture model has been the more preferred technology. For example, the log based data replication technology will generally have less impact on overall database performance when compared with a trigger based implementation.

An issue for the various data replication architectures is that they have mainly been focused on dealing with structured data. The demand for replicating unstructured data has not been resolved by the data replication technologies available today.

3.2.5 Custom build

Traditionally, many organizations started building data copying programs to enable and enrich various systems with data. This was started long before standard tools where available to fulfill the information integration needs of these organizations. These custom build solutions have in many cases been replaced with new standard tools that provides the required low cost sophisticated integration and data movement facilities. But there are still many custom build integration and data movement solutions in production around the world. These implementations do play an important role in the organizations that use them, but as time has passed it has become more and more costly to further develop and maintain the solutions. Organizations today want to have standard solutions that can, in a cost effective way, be integrated and isolated so it is possible to easily replace individual components or entire architectures.

The trend, and the feasibility, of many applications to move towards an SOA enabled environment has applied pressure on these proprietary custom built solutions. The market now wants solutions built on standard software, standard packages, or even built on open source standards which are becoming more and more a business demand.

3.2.6 Combined approach

Companies seldom choose just one architecture or solution to solve their entire integration infrastructure. Because of the often different demands for transformation, volume, velocity, variety and access demands to data, a single architecture will rarely be able to solve all of the integration and data movement needs of an enterprise. So it is quite typical to see an enterprise using ETL tools to solve their ODS and data warehouse data population requirement, and also using either EAI or EII for ad-hoc information and application integration needs.

ETL versus EII versus EAI versus data replication versus custom build

We have chosen to list a set of capabilities we find important to consider when comparing an architecture for a data warehousing environment and a real-time enterprise requirement. The list is not prioritized by importance and does not necessarily represent a full list for qualifying which architecture to choose for which tasks. The comparison of the architectures are described for how the current technologies are most typically used and not with the regards to individual product capabilities.

Data Flow

To resolve a data flow all architectures can be bidirectional, even though traditional ETL has been deployed as one-way data flows. Data replications are by default designed to be bidirectional, but are often also implemented as unidirectional.

Data Movement

ETL offers a scheduled batch oriented approach which compares with both EII running in right-time direct access mode through open access protocols, and EAI which typically offers an asynchronous transaction triggered and managed mode.

Latency

ETL typically has been used with workloads that require hourly-daily-monthly high-latency data availability. This has partly been because of the potential load that ETL can put on a source system, and the historic use of ETL in data warehousing environments. EII can access most sources with zero-latency, and some sources, such as Web services, with low-latency. Whereas, EAI typically only supports message based low-latency access. Data replication also provides a low-latency replication of data through its transaction log or trigger based data capture. The custom-build data movement is typically implemented similar to either an ETL, or a message based architecture, but without the use of the standard ETL engine for both performance and reuse.

Transformation

This is the area where ETL has been very rich on functionality that supports complex data profiling, quality, and transformation operations. EAI can deliver similar capabilities, but they are not as functional as ETL build for transformation and metadata purposes. Both EII and data replication traditionally have not been strong on complex transformations. Therefore, for very complex transformation of data, ETL is today still the best solution.

Event monitoring

Event monitoring is an important capability for a real-time enterprise because it is an excellent mechanism to enable closed-loop functionality. ETL engines have very limited support for event monitoring because of their typically high latency architecture. The EII engine itself also has limited capabilities here as it will depend on trigger functions in the data source. Many data sources for unstructured data do not typically have capabilities to support this. EAI solutions are the best for event monitoring. They enable logic to be added to support true event propagation, in addition to data transaction movement. Data replication can be used to capture data transaction events and support the subsequent data movement.

Metadata

ETL is typically a highly metadata driven process, built around such capabilities as transformation rules and data patterns. All of this is typically described by the metadata repository that is provided with almost any ETL tool. EII typically relies on internal metadata that defines any available sources to the EII environment that are not yet supporting any metadata standards. The same is currently true for most EAI implementations. As described in 3.3.2, “Master data management” on page 95, organizations need to have more of an enterprise-wide metadata approach that is supported across heterogeneous environments.

Workflow
ETL has excellent workflow control capabilities. It is good for scheduling, dependencies, and error and exception handling. Typically EAI environments have extensive rules based capabilities, whereas EII typical does not support any workflow control.

3.3 Other considerations

In this section we look at some of the additional considerations that can impact data access and data movement, even though they do not directly perform the functions themselves.

3.3.1 Database

There are many functions performed by the database, other than simply storing and retrieving data. And the data in the database can be used in many different ways to satisfy the needs of the enterprise. In this section we consider a few of those uses.

Operational systems

Most organizations have many independent sources of data that are often deployed on a number of different hardware and software environments. As a result, they also very often have different database and content stores architectures. This has to be considered when making the choice of an information integration architecture. Most organizations today have a preferred database technology that impacts their decision when purchasing new operational business solutions. The major business application vendors today typically support all the major database technologies.

Data warehousing

The requirements for capabilities in database technology for a data warehousing environment are quite different than those for an operational environment. A data warehouse database, for example, needs to be scalable in terms of loading, support for advanced ad-hoc query, and being able to deliver large result sets. Whereas an operational environment supports many short running transactions with very small storage requirements.

IBM DB2 has stepped up to the challenge of providing techniques and technologies for creating and using large data warehousing environments. Two significant examples are the multidimensional clustering and range partitioning.

Optimizing access using Multidimensional Clustering

Multidimensional Clustering (MDC) was designed to improve performance and reduce table maintenance.

Relational database management systems (RDBMS) have traditionally stored data randomly on storage devices, later relying on row ID (RID) indexes as the means of locating and retrieving that data. Multidimensional clustering technology provides another approach, and performance benefits, to storing, locating, and accessing large volumes of data. That technology is now integrated into DB2, as well as its concurrency and recovery mechanisms.

MDC functionality is transparent to both the user and the application. The only requirement is to use an extension to the CREATE TABLE statement in order to specify the organization of the data. Users can define multidimensional indexes on a set of attributes of a table, and use standard SQL for data manipulation and retrieval. For that reason, existing applications with standard SQL constructs can take advantage of the performance benefits of multidimensional access methods within your data warehouse.

Multidimensional clustering is based on the definition of one or more orthogonal clustering attributes (or expressions) on a table. The table is organized physically by associating records with similar values for the dimension attributes in an extent (or block). DB2 maintains this physical layout efficiently and provides methods of processing database operations for significant performance improvements.

Clustering indexes are used to physically cluster data on insert according to the order the clustering index. Such organization of data and index improves performance of range queries and prefetching. A clustering index is a special type of RID index that contains entries for all existing combinations of the index key column and is used by the insert process to keep the rows of the table in physical sequential order by the values of the index columns.

In data warehousing there are methods to control the insert/delete process to increase the likelihood that there is free space for the clustering ratio to be maintained. The cluster ratio tends to degrade after the free space is exhausted because inserts, updates and deletes may cause the data to be placed out of sequence. Once the clustering ratio drops below a certain threshold, the optimizer no longer considers the table to be clustered. At that point the table must be reorganized.

Range Partitioning

DB2 9 has the capability to support all three common methods of database partitioning. It can simultaneously handle range or table partitioning, multi-dimensional clustering and hashing. That is, you can arrange and order information in the way that best suits your business requirements.

With the ability to use all three partitioning mechanisms together, you can:

- Create and manage much larger database systems by putting multiple table spaces under a single table.
- Divide data across multiple machines.
- Organize data by dimension.
- Manage data load or backup by partition.
- Improve query performance with separated data that prevents scans of irrelevant data.

The result is that you can store tables across more than one tablespace and increase table capacity limit, along with your ability to make larger databases and larger tables.

3.3.2 Master data management

One of the first things that companies discover when trying to reduce process latency is that their most vital data is often stored across many systems, with little or no consistency. This forces development projects to go to extraordinary lengths to reach the correct sources of data and rationalize them into a single semantic representation. In most cases, it also means that applications and users rarely have a complete view of the enterprise data. This is commonly seen in customer marketing and customer service initiatives, where obtaining a single view of the customer still remains an elusive goal.

What they are really seeking is a single view of their master data. And this view takes into account the relationships and dependencies of all the enterprise data in order to support business processes and make vital business decisions. The term for this is master data management.

IBM defines MDM as the set of disciplines, technologies, and solutions used to create and maintain consistent, complete, contextual and accurate business data for all stakeholders. It focuses on the concept of data objects, which represent the key business entities. Core master data objects include such elements as products, organizations, locations, trading partners, employees, customers, consumers, citizens, assets, accounts, and policies.

But is that not what we have been doing all along? Do you have one view into your customers, suppliers, employees, products and materials that accurately reflects the facts?

Unfortunately problems related to the enterprise management of *master data* are more acute than ever before. Creation and transmission of data describing core business entities such as customers, locations and products continues to grow exponentially. No single enterprise business application has become a true *system of record* for master data in the enterprise. At the same time data quality continues to emerge as a legitimate business issue driven by inter-operability and compliance requirements. Adding to the complexity, a new generation of browser based portal style applications drives requirements for more discrete master data sets to be delivered in real-time and quality assured.

What does that mean? It means that at the very time your business requires instant access to high quality information about your core business entities, the facts that describe them are scattered around and in many data transaction structures, databases, data marts, and spreadsheets. And many times in locations you may not even know exist! This is a primary reason why many companies have discovered, as an example, that they have many more customer identification numbers than they have real customers. The lack of a single consistent enterprise view of core business entities severely impacts the enterprise.

So back to MDM. The reason for all the interest in MDM is that it offers the opportunity to:

- Implement a data integration platform that can access the facts about your core business entities from anywhere in your enterprise.
- Automate the creation of a single logically correct view, based on your business rules, that agrees with the facts in the real world.
- Deliver that high quality master data to your current suite of business applications in real-time.

MDM is really a hybrid business application. In one sense it is already included in the current or earlier packaged enterprise applications portfolio. In another it is fundamentally a data integration problem. And that data needs to be available in real-time across applications so they can manipulate the facts about your core business entities in a consistent fashion.

Whether it is reducing cost, meeting legal requirements for compliance, exchanging standard data with partners, consolidating applications, implementing new applications or driving value out of enterprise applications, master data needs to be processed in a consistent fashion. A best practice today suggests a methodology to drive data integration development and related business process alignment, is critical to success. The overall solution framed by the methodology needs to leverage the current or earlier enterprise applications portfolio, and integrate the business applications and integration services required to acquire, transform and deliver master data in real-time throughout the enterprise.

We have been doing this all along. But the opportunity to create a master data service layer in your next generation of IT architecture, that drives benefits for a wide range of critical business initiatives, is worth better understanding now.

Creating a consistent data source

MDM is simply a way to provide consistent, comprehensive core information across an enterprise. But it is more than a simple centralized master data store, it can contain business logic components to manage business rules, event detection and management of privacy and security rules, data validation, and duplicate suspect processing.

An MDM solution also greatly contributes to reduced merger and acquisition costs and time. It challenges us to discuss the usage of information and how it can help to make the creation of a real-time enterprise a reality.

In a typical data warehouse implementation, an ETL process is used to extract source data from operational sources, and apply transformations to make it more meaningful and more easily understood by users. The data is then loaded into the target data warehouse or data marts. Data standardization is also associated to the ETL process in order to guarantee that the same information that exists on different source databases is normalized before loading into the data warehouse. The ultimate goal of the ETL process is to delivery accurate information to the users and guarantee that they have access to the same sources of data.

Business process management is gaining attention in many companies to help enable them to optimize their business processes. Business process management solutions are implemented to automate, monitor, and control operational business processes. Operational decisions related to a process can be fully automated or they can be combined with human activities. The also can interact with business intelligence applications. For example, they could obtain a customer score that is stored on the data warehouse and use it to automate a decision point of a specific process. A business process also can send notifications (alerts) that are related to a state or conditions of an operational process that requires some type of action. Such notification could simply require further investigation and analysis by the decision-maker before actions are performed.

One of the key requirements of the integration of business process management and business intelligence is data integrity, especially with respect to the metadata. This is depicted in Figure 3-7.

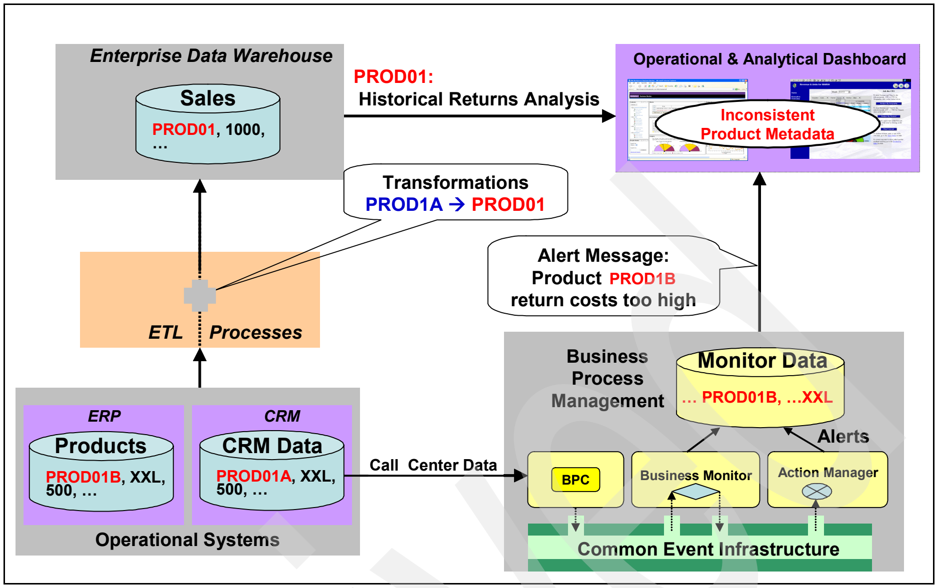


Figure 3-7 Integrating BI and BPM

In this scenario there is no synchronization between metadata in the data warehouse and the metadata in the business process management monitor database. An alert message was sent to the user regarding the high number of returns for a specific product (PROD01B), and will need to perform further analysis, using the data warehouse, to determine the root cause of this issue. Because the data warehouse contains normalized product data, it could be very difficult to relate the product analysis from the data warehouse with that derived from the dashboard - which was derived using data from the process monitor database.

Most companies today have multiple systems that require the same information. Often, each system utilizes its own data versions. And all too often, this leads to misaligned and inconsistent master data. This can lead to costly data redundancy and misleading analytics. MDM is an approach used to control, or eliminate, the proliferation and inconsistency of data.

With an MDM solution, the inconsistent data and misleading analysis between the data sourced from the data warehouse and the operational dashboard sourced from the monitor database potentially could have been avoided.

Because MDM utilizes SOA, it can easily be integrated with the process. And it also can be integrated with the ETL processes for the population of the data warehouse. As represented in Figure 3-8, MDM is a primary source of product information that enables the synchronization between operational analyses and data warehouse analyses. A similar approach can be used to synchronize customer analyses that are sourced from both systems.

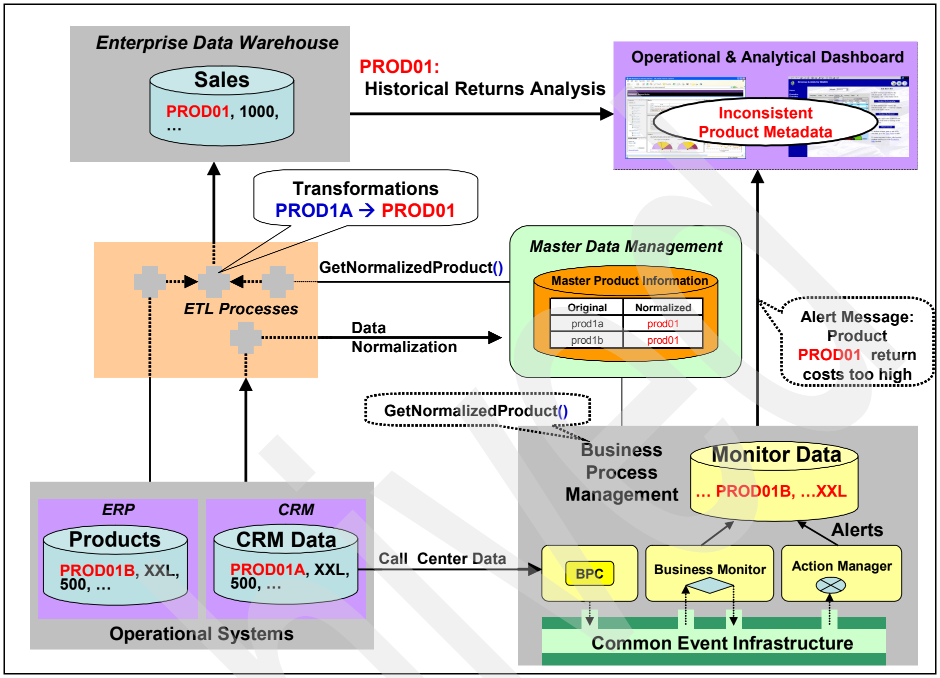


Figure 3-8 MDM and analytics

3.3.3 Analytic applications

Analytic applications are not a new thing. They have been around in one form or another for a number of years. However, now that we are evolving further into the realm of data warehousing and business intelligence the time has come to dust them off, enhance them, automate them, and use them as another critical capability in the world of analytics, or business intelligence as we prefer.

For many years we have asked people to dive into the depths of the data in a data warehouse, analyze it, and use the results to resolve all the issues of the enterprise. Unfortunately that has not happened, at least to the degree which we would have liked. And the likelihood is that it will not happen, because there is just too much data and too few people with the domain knowledge required to really understand and analyze the data. So, we need help.

Analytic applications to the rescue. What we an do is glean some of that domain knowledge from the gifted few, and instantiate it in the analytic application. And what does not get put into the application, should find its way into a rules base.

What we are doing is leveraging the domain knowledge and, where applicable, enabling it to be automated. Here is a way to enhanced efficiency and effectiveness.

With this domain knowledge and rules base, an analytic application cannot only access data, it can analyze it and take action - even if that action is as simple as sending an alert to a human. Or, we can further extend that application capability to enable the application to better equip the human. That is, with embedded knowledge it can help guide the human. Rather than simply alerting the human, the application can offer plans of action. For example, it could recommend looking at a particular area of the data warehouse, or other related data.

Basically what you are doing is embedding the knowledge and problem solving capability of a number of skilled analyst into the analytic application. Although not perfect, it is a giant step in the right direction. It is enhanced BI in a real-time environment. Here is another characteristic of a real-time enterprise?

In-line Analytics

Today, business intelligence applications are used in a relatively small percentage of corporate environments. This is particularly true when considering the front-line workers who are executing the business processes. One of the primary reasons is that business intelligence applications may be more difficult to use because they need data at the right time and in the right context to enable them to make decisions, and many times there are also complex analytics involved. Therefore it is critical that businesses embed business intelligence tightly into the business processes - in the form of in-line analytics in their applications.

In-line analytics represent the ability to provide analytical services at the point of decision, to optimize decision making. This means that analytics need to be an integral part of the business process. As a user or system is at a point of decision within the business process, analytics are delivered in the appropriate business context.

Types of in-line analytics

In this section we discuss three ways for the delivery of in-line analytics:

1. **System-driven analytics** refer to a system that programmatically consumes an analytical service for decision-making. That is, decisions are made using information as a service. As an example, consider a customer who wants to place an order. During the process, the system will programmatically suggest payment methods, using a data mining solution, and based on a score from the customer payment history.
2. **User-based** real-time analytics are analytics provided within a process step that requires user intervention. It is critical that the decision is made in

real-time. As an example, a customer calls into the call center and requests an exception on the return policy. The call center agent is provided with real-time information about the customers historical purchasing information to help determine if the exception should be granted.

3. **User-based complex analytics, and guided analysis** can be time-consuming. For example, they often require use of OLAP to analyze the problem across multiple dimensions, or analysis across multiple data sources as they search for appropriate data.

   For example, assume there are an unusually high number of returns for a particular product. To find the root cause, the analyst will have to analyze the problem across different dimensions. A skilled and experienced analyst can do this. But, to enable other less skilled workers to perform these analyses requires some help or additional tools.

   One such type of help, or tool, is called guided analysis. In simple terms, this is an application that has embedded in it the analysis techniques of a skilled analyst. As such, it can provide suggestions and analysis paths to help guide the less skilled worker to the root-cause problem. For example, it can provide information about any similar past problems, suggest appropriate data sources to analyze, and specifics on what they should be looking for.

Guided analysis applications can be very valuable in enabling a wider range of workers, rather than only to more highly paid analysts, to perform problem analysis. As such, problems can be resolved more quickly and for much less cost.

Dashboards

In discussing analytics, it is important to include dashboards. This is one key way to visualize the need for, or the result of, analytics. They are a primary output mechanism for the analytics.

Dashboards are intuitive, informative and possibly interactive displays. For an example, refer to Figure 3-9. They include gauges and meters that enable visual display of information for fast and easy understanding. For example, this is where alerts can appear to enable fast action to be taken. Visually noting that a key performance indicator is approaching a threshold value can enable action to be taken before a problem occurs or becomes critical.

Dashboards are discussed in more detail in 4.2, “DB2 Alphablox” on page 148.

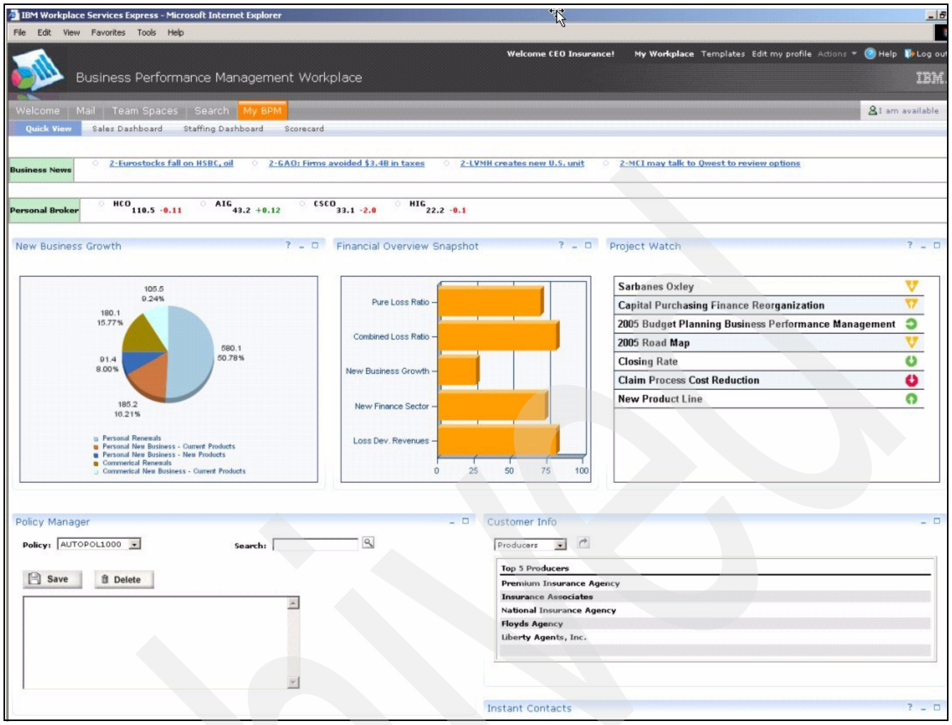


Figure 3-9 Dashboard example

3.3.4 Closed-loop environments

The notion of closed-loop is, together with analytic applications, the base of the real-time enterprise.

Transaction data is captured and integrated into the data warehouse. Based on key performance indicator thresholds and user defined triggers, that data can be can fed to analytic applications. The output from those applications, in turn, is the directly fed back to the front-line business decision makers in the form of recommended actions. This can be performed automatically by, for example, dynamically making price changes for specific items. Or it can involve manual activity by providing, for example, guided analysis for understanding, analyzing, and resolving problems.

These types of closed-loop capabilities can create a zero-latency, or low-latency, environment. That is, a faster time to problem identification and resolution, and thus a proactive avoidance of problems. Or, at least, a minimization of the impact of those problems. A zero-latency environment allows the company to integrate analytics into day-to-day business operations and shorten the time between business decision and business action. To implement such a zero-latency system, the closed-loop process would typically need to be automated. And such a system would require a real-time data warehousing environment, be integrated with a real-time analysis engine, and leverage a real-time rules-based engine (a decision engine). The closed-loop environment would dynamically adjust the business operations based on messages generated by the decision engine.

This scenario describes the type of environment for which we strive in a real-time enterprise. It does not mean that every process in the enterprise is developed in this manner, rather only those where it is required, achievable, and cost effective.

3.4 Information as a service

An important topic in the IT industry today is that of providing information as a service. It begins to further standardize the approach for information access and interchange, particularly in a heterogeneous environment. In addition, it can significantly reduce the cost and time for application development. These are key advantages as we move forward in implementing a real-time enterprise environment. But, there must be standards and formats defined for guidance, flexibility, and stability as we go forward. And, there are.

In this chapter we describe various architectures used in a real-time enterprise environment. Each of the mentioned areas, architectures, and products in the market, today offer extensive, or at least some, capabilities for providing information as a service. One specific example is SOA, the acronym for a service oriented architecture.

In this section we provide a brief overview of services and a service oriented architecture. A service can be defined as any discrete function that can be made available to a client. These functions, or services, are reusable and application independent. Thus the benefit of reduced cost for application development.

To be usable and interchangeable a service should be based on standards with clearly defined specifications for interaction, such as format, function, and exception handling. These definitions may be handled by means of the Interface Definition Language (IDL) or the Web Services Definition Language (WSDL).

3.4.1 Service Oriented Architecture

From a non-technical perspective, SOA is a set of business, process, organizational, and governance methods that help to create an efficient, effective, and agile business environment. From a more technical perspective, it is a way to standardize and improve application development and execution through the use of easily accessible, standardized, and reusable services. These services are independent of hardware and operating environments.

Most companies today are being pressured by their customers and shareholders to drive growth by improving productivity and limiting cost in every aspect of their operations. But how can companies do that if they have rigid, expensive, and proprietary IT systems. It is a difficult task indeed. One of the most value things a company can do today is implement systems with ***flexibility***. That is, flexibility for such things as meeting new market demands and seizing opportunities before they are lost. To increase flexibility, a company has to look at its business as a collection of interconnected functions, or discrete processes, such as checking customer credit or authenticating a user. Then they can decide which of those functions are core, or differentiating, and which can be streamlined, outsourced, or even eliminated. If the company can mix and match these functions at will, or dynamically, in response to changing business conditions, they will gain a significant business advantage. But to achieve this degree of flexibility in the business operations, the company will need an equally flexible IT environment. One way to do this is through a SOA.

SOA is also an applications framework that makes it easy to ***reuse*** and combine the discrete business processes defined for the business. It is comprised of individual functional components that can be arranged and rearranged. With a SOA, the company can build, deploy, and integrate applications and link heterogeneous system and platform together across the organization.

IBM has developed a SOA reference architecture that provides a flexible, extensible, open-standards-based infrastructure. This is a key added value of the IBM solution. The architecture specifies the required capabilities and services, and the defined interfaces that enable integration of the solution components. This is graphically depicted in Figure 3-10).

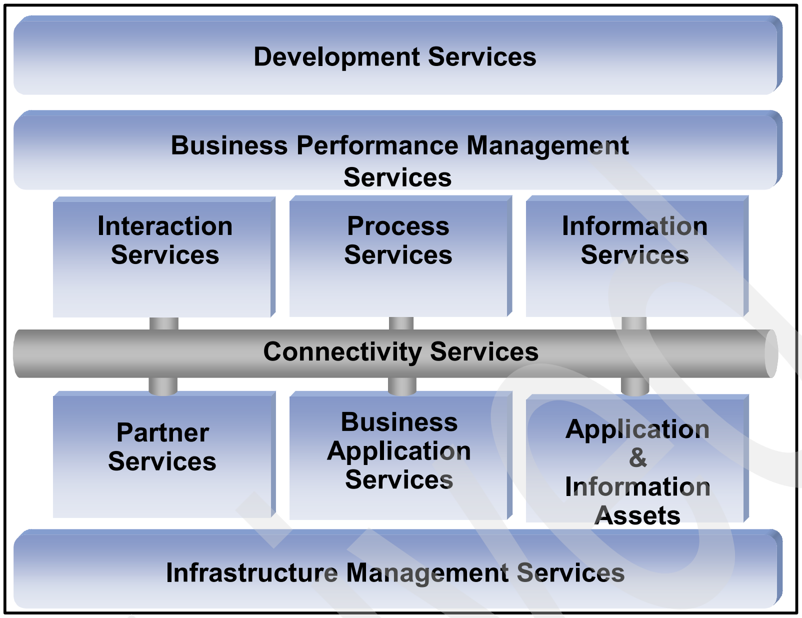


Figure 3-10 IBM SOA reference architecture

We have indicated that benefits of SOA are flexibility and component reuse. But, what are the forces driving this need for increased flexibility? Well, for one thing consider economics. As the marketplace continues to globalize, new markets, new workforces, and new competitors are forcing companies become more flexible and able to adapt more quickly.

To support this, we see the cycle time between changes in business and business processes continually getting smaller. While you might have seen companies make significant changes yearly in the past, you will now begin to see the same level of change on a quarterly, monthly, weekly or even a daily basis.

While business leaders were focused more on cost-containment in the past, we are seeing that growth is now back at the top of the agenda today for the CEO. And, that growth demands flexibility so you can be more nimble than your competitors.

This is not to say that cost reduction has lost its importance. On the contrary, businesses are looking even harder for ways to make better use of their investments. There is more information available today than ever before, and companies need to be able to analyze it regardless of its location, format, or type.

And finally, SOA and the flexibility it brings is crucial for becoming an on demand real-time enterprise. SOA blends the best of all these concepts. But it is important to recognize that SOA is not the end of the road. It is the next step in the evolution of flexible infrastructures, that can help you get much further down the road - and much more quickly.

3.4.2 Enterprise service bus (ESB)

At the core of SOA is the enterprise service bus (ESB) which enables organizations to make use of a comprehensive, flexible and consistent approach to integration. Due to the complex and varying nature of business needs, an ESB is essentially an architectural pattern which unifies message oriented, event driven and service oriented approaches to integration. The functionality of the ESB is delivered in the IBM WebSphere Enterprise Service Bus product.

An enterprise service bus provides an infrastructure that actually has no direct connection between service consumers and providers. Both are connected to the bus, but not directly to each other. So, the consumers connect to the bus, but not the provider that actually delivers the service. Therefore, there needs to be standards, or defined formats, for the interactions of finding, delivering, and using the service. And, this must include the ability to provide security and delivery assurance that is outside the actual application. Here is another benefit of the ESB. And that is further reduced costs, and faster application development.

An ESB can optimize the delivery of information and services to help improve cycle time, reduce costs and improve IT flexibility and connects an infrastructure to integrate applications and services, which can power the service oriented architecture. By moving to a Web services based architecture we will de-couple implementation-specific interfaces from their applications, and reduce the impact of changes to either the application logic, or the implementation of the interfaces.

An ESB further shrinks the number of interfaces, enabling more flexible coupling and de-coupling of the applications, and improve the potential for reuse of both the application logic, now presented as application services, and the connectivity and integration logic defined in the ESB.

By deploying an ESB as the connectivity and integration infrastructure within a business or organization, it should enable faster, simpler and more flexible changes to the organization, helping to reduce up-front investments and drive down maintenance costs.

Most infrastructure technologies are today capable of acting in a service enabled environment. However not many of the enterprise applications that are in the market today are ready to implement purely based on a loosely-coupled service environment. Some applications still have a vital need to be close-coupled to certain information stores, and databases. An example of this is implementation of an ETL process that has to extract large amounts of data, transform it, and load it into a data warehouse, These are typically today still close-coupled to both the transactional database and the data warehouse database because of the performance profile of the processes.

An enterprise service bus example

In this section we describe an environment that has many disparate technologies and architectures, that are all built around a SOA. To implement such an environment, we use an enterprise service bus architecture.

In Figure 3-11, we have XML data from a customer or partner coming in via EAI. It could be data that has been processed in an ERP or CRM application, and is put on the ESB to become available to other applications and systems. We also have a collection, or set, of distributed objects coming in via EII. This could be, as an example, a partner contract retrieved from a common content repository, it could be product manuals or documentation, or it could simply be access to operational customer data.

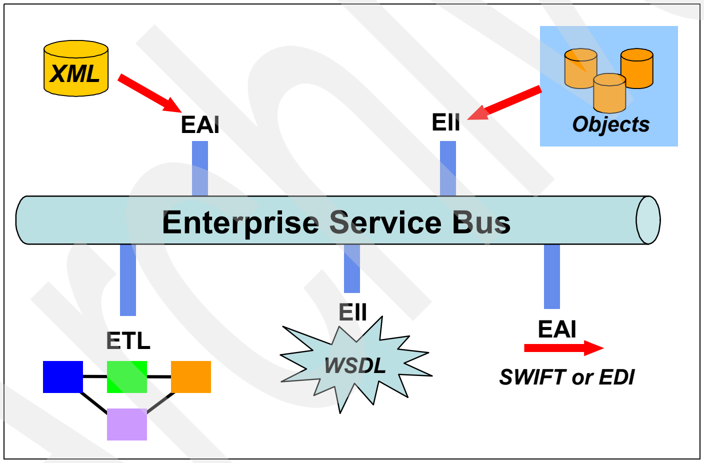


Figure 3-11 Enterprise service bus

Then we have an ETL process that can be the cleansing and transformation service that is used to reconcile or validate data from any of the other services in the infrastructure. We can also see this as the service that will be used to populate the enterprise data warehouse in both a low-latency and a batch oriented architecture. Then we have an example of EII that publishes a Web service for partners to access sales data specific to their deliveries to our organization. Or it could be the opposite, where this is the integration to a combination of different public services being made available internally. And lastly, we have an EAI service that manages the Society for Worldwide Interbank Financial Telecommunication (SWIFT) or electronic data interchange (EDI) transactions to be prepared for an electronic integration with business partners.

3.4.3 ESB and the real-time enterprise

The ESB architecture is key to the real-time enterprise implementation. Most of the back-end technologies are already services enabled, and will be capable of operating in an ESB environment.

In Figure 3-12 we illustrate an example of a closed-loop scenario with an ESB. There is an ETL engine that is responsible for extracting data from the transactional systems. The ETL engine is then responsible for transforming, preparing, and loading the data into the data warehouse depicted. Its assumed in this example that the data warehouse does not have a need to provide access to data with zero-latency, and therefore it is only updated on a daily basis.

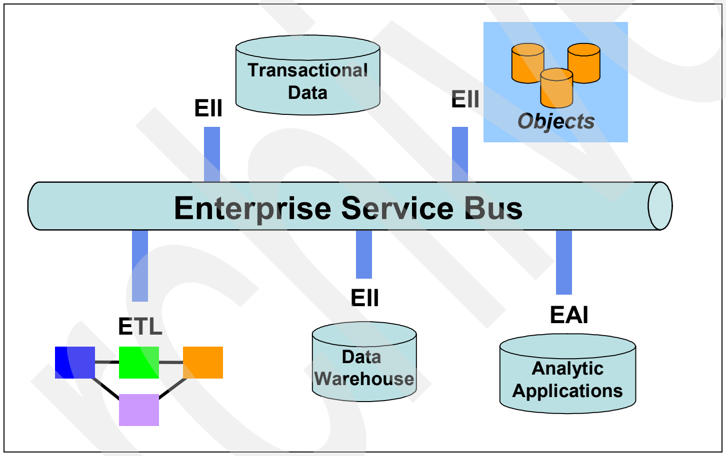


Figure 3-12 Example closed-loop scenario with an ESB

However, there is an analytic application (a data mining engine for example) that in our example is used for customer segmentation to verify if certain offerings should be provided to specific customers at a given point in time. The data mining engine requests data from the data warehouse, but could also have a need to look to the EII engine to locate zero-latency transactional data from internal customer transaction data, or look for external market trends, or even request new segmentation mining models.

In this example, the data mining application is capable of delivering customer segmentation, or rating results, back to the operational transaction system based on both the historic data warehouse, and the operational and external market trend data. The transactional system can then, with this customer segmentation and rating information, decide how a specific customer situation should be processed. And this has all taken place in the real-time enterprise.

In this chapter we have discussed and described some of the architectural considerations for developing a real-time enterprise. We have also discussed techniques and technologies that can be used to do so, along with their capabilities. In the remaining chapters of this redbook, we provide additional technology information, implementation approaches, and a case study example to further assist in understanding a real-time enterprise environment.

4

IBM technologies supporting real-time

In this chapter we introduce you to several IBM technologies and products that can be used when developing your real-time enterprise. A number of these products were used in this redbook project environment. We discuss how they interact, and their role in a business intelligence environment.

The technologies on which we focus are:

- DB2 Data Warehouse Edition (data)
- WebSphere Information Integration (federation and integration)
- Process Integration (process modeling and monitoring)
- WebSphere Message Broker (application connectivity)
- WebSphere Enterprise Service Bus (predominantly Web services-based environments)
- WebSphere MQ (Application Connectivity)
- Balanced Configuration Unit

To focus on these solutions, we have organized them according to the IBM brand and category positioning.

4.1 DB2 Data Warehouse Edition

Business intelligence (BI) is a critical business advantage. Most businesses have some level of BI capability, but must continue to grow their capabilities to remain competitive. DB2 Data Warehouse Edition is a powerful set of products that rapidly enables enterprises to develop robust BI solutions, such as those with data mining techniques, in-line analytics and an integrated design tool.

The DB2 Data Warehouse Edition (DWE) is a suite of products that combine the strength of DB2 Universal Database™ (DB2 UDB) with the powerful business intelligence infrastructure from IBM. DWE provides a comprehensive business intelligence platform that contains products with functionality to build and deploy a robust business intelligence solution.

With the DWE, you can build and deploy a data warehousing environment that includes a highly scalable relational database, integrated information gathered from numerous heterogeneous data sources, business intelligence analytics, and front-end financial analysis tools.

> **Note:** This redbook is being published at a time when DWE is evolving. That is, the existing version is DWE V8.2, but DWE V9.1 has just become available. As it is a time of transition, we have used both versions at times. Providing information about both versions offer a degree of continuity in understanding.

With DWE V8.2, there are three editions available to satisfy the requirements of any particular business enterprise. We now take a look at those.

4.1.1 Data Warehouse Edition V8.2

DWE V8.2 includes DB2, federated data access, data partitioning, integrated online analytical processing (OLAP), advanced data mining, enhanced extract, transform, and load (ETL), workload management, and spreadsheet integrated BI for the desktop. DWE works with and enhances the performance of advanced desktop OLAP tools such as DB2 OLAP Server™, and others from IBM partners. Specifically, this version includes:

- DB2 UDB Enterprise Server Edition, V8.2
- DB2 UDB Database Partitioning Feature, V8.2
- DB2 Cube Views™, V8.2
- DWE Intelligent Miner™ Modeling, Scoring and Visualization, V8.2
- DB2 Office Connect Enterprise Web Edition, V4.0
- DB2 Query Patroller, V8.2
- DB2 Warehouse Manager Standard Edition, V8.2
- WebSphere Information Integrator Standard Edition, V8.2

DB2 Data Warehouse Standard Edition, V8.2, is a complete data warehouse infrastructure package that includes DB2, integrated OLAP, advanced data mining, ETL, and spreadsheet integrated BI for the desktop. DWE works with and enhances the performance of advanced desktop OLAP tools such as DB2 OLAP Server and others from IBM partners. Specifically, it includes:

- DB2 UDB Workgroup Server Unlimited Edition, V8.2
- DB2 Cube Views, V8.2
- DB2 Intelligent Miner Modeling, Scoring and Visualization, V8.2
- DB2 Office Connect Professional Web Edition, V4.0

DB2 Data Warehouse Base Edition, V8.2 provides mid- to large-scale enterprises with a data warehouse and data mart infrastructure that includes DB2 and integrated OLAP capability that can support departmental warehouses or lines of business, or enterprise data warehouses where there is a need for scalability and performance. This edition includes:

- DB2 UDB Enterprise Server Edition, V8.2
- DB2 Cube Views, V8.2

In the following sections, we provide a brief description of the components of the DB2 data warehouse editions.

DB2 UDB Enterprise Server Edition

DB2 UDB Enterprise Server Edition is a highly scalable, multi-user relational database management system that can manage data warehousing, online analytical processing (OLAP), online transaction processing (OLTP), and data mining.

DB2 UDB Database Partitioning Feature

DB2 UDB Database Partitioning Features allows you to create and manage partitioned databases. Partitioned database systems can manage high volumes of data with near-linear scalability while providing benefits such as increased performance and high availability. The data from a particular database is partitioned across a number of database instances, as depicted in Figure 4-1 on page 114.

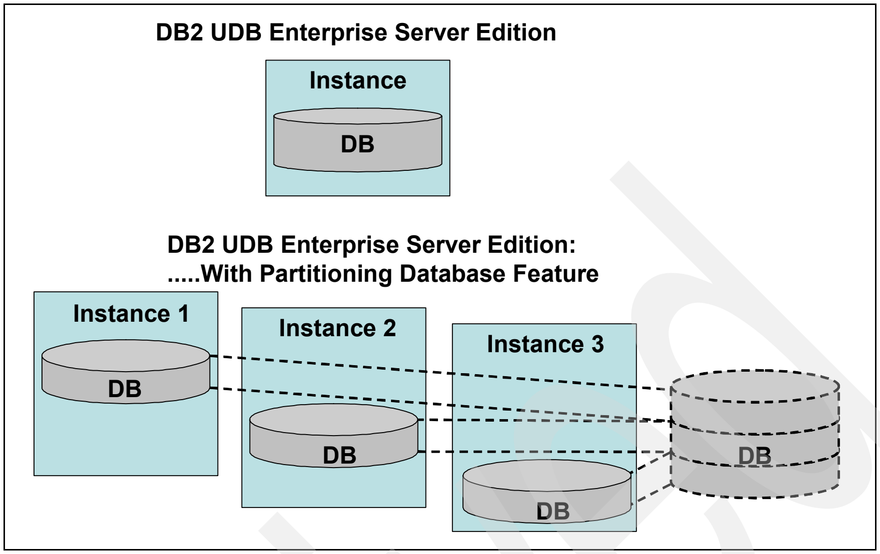


Figure 4-1 DB2 Enterprise Server Edition and Data Partitioning Feature

DB2 Cube Views

DB2 Cube Views benefits business intelligence users by decreasing the development time to define shared OLAP models across multiple tools. With new metadata objects and tooling, you can define Cube Views after the metadata is imported to the DB2 UDB catalog. This metadata can also be used in other business intelligence tools. Since the shared common metadata includes aggregation formulas and calculations, you can benefit from greater consistency of analytical results across the enterprise.

DB2 DWE includes features that are designed specifically to improve the usability and performance of business partner business intelligence products. Data warehouse administrators can build optimal structures in DB2 UDB based on the recommendations of the DB2 Cube Views Optimization Advisor, which can improve the performance of OLAP queries that are generated by partner tools. By using these business analysis tools with DB2 UDB managed tools, you receive more consistent query results. Designers can easily import cube views and mappings from DB2 UDB rather than define the same metadata separately in other partner tools that might be used in the same enterprise.

DB2 Cube Views works with IBM DB2 Office Connect Analytics Edition, a Microsoft Excel® plug-in, to display and navigate Cube Views in DB2 UDB. IBM DB2 Office Connect Analytics Edition turns your Excel spreadsheet into an OLAP tool. For more information about other editions of DB2 Office Connect, see “DB2 Office Connect Enterprise Web Edition” on page 116.

You may see the phrase DB2 Multidimensional Metadata Management when reading about products. That phrase refers to DB2 Cube Views.

DB2 Intelligent Miner Modeling, Scoring and Visualization

With DB2 Intelligent Miner, you can embed mining analytics in business intelligence, e-commerce, or traditional OLTP application programs.

DB2 Intelligent Miner includes the following components

- Intelligent Miner Scoring
- Intelligent Miner Modeling
- Intelligent Miner Visualization

Intelligent Miner Scoring

Intelligent Miner Scoring enables applications to apply predictive model markup language (PMML) models to large databases, a subset of databases or single rows or cases. The PMML models can be created by one of the Intelligent Miner products or by other applications and tools that support PMML models.

Scoring uses existing data mining models and simply applies these models to new data. For example, you might have a classification model that contains rules about estimating the churn risk for customers. Given the profile data about a particular customer, the scoring function computes the churn risk. You can then apply this function in real-time to a single record. For example, the customer who is currently talking to someone in the call center.

Intelligent Miner Modeling

To understand underlying processes, or behavior of your enterprise, you can build mining models from your enterprise data. For example, an association model describes consumer behavior. A predictive model is an equation or set of rules that makes it possible to predict an unseen or unmeasured value from other, known values. You can view models with Intelligent Miner Visualization.

Intelligent Miner Modeling is a process in which you make calls to the DB2 UDB data mining functions. You can base your models on data mining algorithms such as associations discovery, clustering, or classification.

Intelligent Miner Visualization

Intelligent Miner Visualization provides Java™ visualizers to show data models in a graphical format. Applications can call the visualizers to present model results, or you can deploy the visualizers as applets in a Web browser.

Visualization provides analysts with visual summaries of data from a database. It can also be used as a method for understanding the information that is extracted by using other data mining methods. Features that are difficult to detect by scanning rows and columns in databases often become obvious when you view them graphically.

DB2 Office Connect Enterprise Web Edition

With DB2 Office Connect, you can connect your Excel spreadsheets to databases so that you can retrieve, modify, and update the data in your database.

The DB2 Office Connect client, which you can access after you create a DB2 Office Connect project, turns Excel into a client of a DB2 Office Connect repository. You can use the client to retrieve the data from DB2 Office Connect projects and perform database operations.

DB2 Query Patroller

Although DB2 Query Patroller is bundled with the DB2 Warehouse Edition V8.2, its implementation, use, and administration, will be changed in V9.1.

DB2 Query Patroller traps queries from enabled DB2 UDB clients, analyzes the queries, and then dynamically prioritizes and schedules them for processing.

- You can use DB2 Query Patroller to dynamically control the flow of queries to your DB2 UDB database. You can also:
  - Define separate query classes for queries of different sizes to better share system resources among queries and to prevent smaller queries from being stalled behind larger ones
  - Give high priority to queries that are submitted by certain users so that these queries run sooner
  - Automatically hold large queries so that they can be canceled or scheduled to run during off-peak hours
  - Track and cancel runaway queries

With DB2 Query Patroller, you can regulate the query workload so that small queries and high-priority queries can run promptly, and you can ensure that your system resources are used efficiently. In addition, you can collect and analyze information about completed queries to determine trends across queries, heavy users, and frequently used tables and indexes.

DB2 Warehouse Manager Standard Edition

DB2 Warehouse Edition V8.2, its implementation, use, and administration, will be changed in V9.1. Included in V8.2 are the following components.

- **Data Warehouse ETL tools:** these are components that enhance the data warehousing extract, transform, and load (ETL) capabilities provided by the DB2 Data Warehouse Center.
- **Warehouse transformers:** these are stored procedures and user-defined functions that can be used to transform data in a warehouse ETL step to clean, invert, and pivot data, generate primary keys and period tables, and calculate various statistics.
- **Warehouse ETL agents:** these agents manage the flow of data between the data sources and the target data warehouses. They use open database connectivity (ODBC) drivers or DB2 CLI (call-level interface) to communicate with the different data sources.

WebSphere Information Integrator Standard Edition

WebSphere Information Integrator, its implementation, use, and administration will be changed in V9.1.

DWE includes a limited-use license of IBM WebSphere Information Integrator Standard Edition, which enables DB2 relational wrappers to be used to access heterogeneous RDBMS sources. These ETL sources include Oracle, SQL Server, Teradata, and Sybase.

WebSphere Information Integrator contains wrappers to enable access to data sources. They are:

- **Relational wrappers:** these are required to access data that is stored in heterogeneous relational databases such as Oracle, Sybase, Microsoft SQL Server, and Teradata. Access to data stored in IBM databases, such as DB2 UDB and Informix, is built into DB2 UDB for UNIX® and Windows®.
- **Non-relational wrappers:** these are used to access non-relational data sources such as table-structured files, Excel files, Extensible Markup Language (XML) documents, BLAST search algorithms, Documentum data, Entrez sources, HMMER sources, BioRS, and Extended Search sources. Through the extended search sources you can get access to a wide variety of unstructured data sources, such as Domino®, Microsoft Exchange, Microsoft Index Server, and Lightweight Directory Access Protocol (LDAP) directories.

4.1.2 Data Warehouse Edition V9.1

DB2 Data Warehouse Edition (DWE) V9.1 is an integrated platform for developing business intelligence solutions. Included are data warehouse-based analytics, including Web-based applications with embedded data mining and multi-dimensional Online Analytical Processing (OLAP). DWE integrates core components and functionality for data warehouse administration, data mining, OLAP and inline analytics/reporting. These platform pillars are based on technology from Eclipse, Rational Data Architect, the SQL Warehousing Tool, DB2 Intelligent Miner, DB2 Cube Views, and DB2 Alphablox.

This is depicted in a functional view in Figure 4-2.

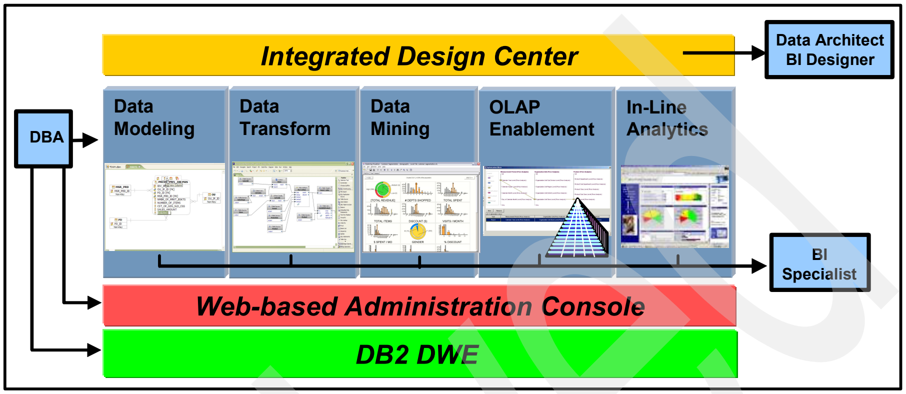


Figure 4-2 DWE Architecture

In DWE Design Studio, physical data modeling, cube modeling, data mining modeling, and SQL data flow/control modeling are unified in one common design environment. That Eclipse-based design environment integrates specific DWE components within a common framework and user interface. Alphablox currently uses a native interface, but there are plans to support the Eclipse plug-in architecture in a later release.

DWE is a component-based architecture with client and server functions, both leveraging emerging IBM Software Group frameworks and shared sub-components. DB2 is the foundation for DWE, providing a scalable data warehousing platform. DWE then extends DB2 data warehousing with analytic tooling (design-side) and infrastructure (runtime), including WebSphere Application Server and Rational Data Architect. DB2 Alphablox is the tool for developing custom applications with an embedded analytics-based dashboard. DWE gives customers faster time-to-value for enterprise analytics, while limiting the number of vendors, tools, skill-sets and licenses required.

DWE OLAP works together with DB2 Alphablox to accelerate OLAP queries. It uses multi-dimensional models to design runtime DB2 objects containing critical dimensions and levels - or slices - of the cube. These pre-joined and pre-aggregated Materialized Query Tables (MQTs) are exploited by the DB2 optimizer, which re-writes incoming queries, transparently routing eligible queries to the appropriate MQT for significantly faster query performance.

Besides performance benefits, DWE OLAP metadata and tooling allow cube models to be defined once in DB2 and also used by Alphablox. Either way, the productivity of OLAP administrators and developers is improved. Because the shared common metadata includes aggregation formulas and calculations, users benefit from greater consistency of analytical results across the enterprise.

Embedded data mining capability in DWE Mining uses algorithms to analyze data warehouse data, in place, and provide insights into business behaviors that are otherwise unknown, invisible and impossible to discover. Data mining in the data warehouse enables improved accuracy and timeliness. In the past, data mining was invoked by an SQL programming interface. In the DWE Design Studio, a new data discovery function allows you to profile your data, sample and view table contents and visualize correlated statistics to understand what parts of the data warehouse hold the best potential for data mining. Next, the Eclipse data flow editor is used to visually design data mining flows with modeling, scoring and visualization operators. Then, SQL can be generated and made available for Alphablox, or any customer application, to invoke the data mining flow for embedded analytics.

Alphablox lets you quickly and easily build custom Web-based OLAP-style reporting either as standalone applications or embedded in Web portals, dashboards and other existing applications. Time-to-value comes from leveraging pre-built libraries of J2EE™ components - the *building blox* of this architecture. Because of the focus on customization and embedding, Alphablox enables applications with embedded analytics to invoke the data warehouse-based analytic structures (data mining and OLAP) modeled and maintained in DWE. Alphablox is described in more detail in “DB2 Alphablox” on page 148.

The new SQL Warehousing Tool automatically generates DB2-optimized SQL based on a visual data flow modeled in the Design Studio canvas, drawing from a palette of predefined operators. The library of SQL operators covers the types of transformations typically needed to populate analytic structures involved in data mining and OLAP, or for any in-database data flows. Data flows can be combined in sequences as control flows which are then scheduled for execution. Since the data flows are SQL-based, DB2 acts as the runtime engine, with WebSphere Application Server providing control and scheduling.

In summary, the new Design Studio in DWE V9.1 provides an integrated platform for modeling, designing, and maintaining data warehouse-based analytic structures, which can be invoked by Alphablox for embedded enterprise analytics.

DB2 Data Warehouse Edition is available in two versions:

- DB2 Data Warehouse Base Edition

- DB2 Data Warehouse Enterprise Edition

A brief comparison of the components in the two DB2 Data Warehouse Editions is provided in Table 4-1.

Table 4-1 DB2 Data Warehouse Editions - components

Base	Enterprise	Component
Yes	Yes	DB2 UDB Enterprise Server Edition
Yes	Yes	DWE OLAP
Yes	Yes	DWE Design Studio
Yes	Yes	DWE Integrated Installer
No	Yes	DB2 Data Partitioning Feature
No	Yes	DB2 Query Patroller
No	Yes	DWE Mining
No	Yes	DB2 Alphablox
No	Yes	DB2 Admin Console
No	Yes	DB2 Design Studio
No	Yes	DWE SQL Warehousing Tool

DB2 UDB ESE is the most comprehensive edition, and is designed to meet the relational database server needs of mid- to large-size businesses. It can be deployed on Linux®, UNIX, or Windows servers of any size, from one CPU to many CPUs. DB2 ESE is an ideal foundation for building on demand enterprise-wide solutions, such as large data warehouses of multiple terabyte capacity, or high performing 24x7 available high volume transaction processing business, or Web-based, solutions. It is the database edition of choice for industry-leading enterprise solutions.

DB2 ESE is available on all supported versions of UNIX (AIX®, Solaris™, and HP-UX - including HP-IA64), Linux, Windows NT® (SP6 or later), Windows 2000 (SP2 or later), and Windows Server® 2003. It does not run on Windows XP for production purposes, but can be licensed for user acceptance testing, test, and application development on this platform (this restriction is in accordance with Microsoft's direction for this operating system and therefore applications running on Windows XP can be adequately serviced by DB2 Express, DB2 WSE, or DB2 WSUE servers).

Connectivity to IBM eServer™ zSeries®-based and iSeries™-based data is provided by the DB2 Connect™ component, and is ideal for certain data replication scenarios and remote administration.

DWE OLAP

With DWE OLAP, no extensive knowledge of OLAP is required to accelerate real-time data analysis. Database administrators (DBAs) can drag multiple objects onto predefined layouts to quickly add OLAP function to the data warehouse. And they can use DWE OLAP to create summary tables and metadata to enable faster data access by business partner tools. DBAs can also develop associated SQL queries so users can start with the summarizations of the cube and then drill down into more customized detail.

DWE OLAP:

- Accelerates DB2 UDB queries by recommending MQTs.
- Enables the DB2 UDB optimizer to rewrite incoming queries to access MQTs.
- Loads cubes, performs drill-through queries and ad-hoc analysis directly, using the relational tables in DB2 UDB.
- Enhances all queries that use aggregate data.

DWE OLAP includes features and functions that transform your IBM DB2 Universal Database system into a platform for managing and deploying multidimensional data across the enterprise. You can create a set of metadata objects to dimensionally model your relational data and OLAP structures. It stores the metadata objects that you create in the DB2 UDB catalog.

DWE OLAP also provides an SQL-based and XML-based application programming interface (API) for OLAP tools and application developers. Through CLI, ODBC, or JDBC™ connections, or by using embedded SQL to DB2 UDB, applications and tools can use a single stored procedure to create, modify and retrieve metadata objects.

In addition, DWE OLAP includes the OLAP Center with which you can create, manipulate, import or export cube models, cubes and other metadata objects for use in OLAP tools. The OLAP Center provides easy-to-use wizards and windows to help you work with your metadata. For example, the Optimization Advisor analyzes your metadata and recommends how to build summary tables that store and index aggregated data for your OLAP-style SQL queries.

The new multidimensional metadata in DB2 UDB provides two major benefits:

- Improves the flow of the multidimensional metadata between business intelligence tools and applications. For example, you can use a graphical interface that is provided to store the multidimensional metadata as part of the

DB2 database and make that metadata available for all warehousing tools and business intelligence applications.

- Enhances the performance of OLAP-style queries. Based on the multidimensional metadata, you can create DB2 summary tables using the recommendations from the Optimization Advisor. The summary tables contain precalculated data that maps to your OLAP structures. Queries that are generated from the warehousing or business intelligence application with the same OLAP structure will gain performance improvement.

DWE OLAP exploits DB2 features such as summary tables, different index schemes, OLAP-style operators, and aggregation functions. The following components are provided:

- Multidimensional metadata objects: You can create a set of metadata objects to dimensionally model your relational data and OLAP structures. DWE OLAP stores each of the metadata objects that you create in the DB2 catalog.
- OLAP Center: With the OLAP Center, you can create, manipulate, import, or export cube models, cubes, and other metadata objects to be used in OLAP tools. For example, the Optimization Advisor analyzes your metadata objects and recommends how to build summary tables that store and index aggregated data for your OLAP-style SQL queries.
- Multidimensional Services: DWE OLAP provides an SQL-based and XML-based application programming interface (API) for OLAP tools and application developers. Using CLI, ODBC, or JDBC connections, or by using embedded SQL to DB2 UDB, applications and tools can use a single stored procedure to create, modify, and retrieve metadata objects.

You can exchange metadata objects between the DB2 catalog and OLAP tools. To import or export metadata objects to or from the DB2 catalog, utilities called metadata bridges are available for specific OLAP and database tools.

DWE OLAP metadata objects describe relational tables as OLAP structures, but these metadata objects are different from traditional OLAP objects. Metadata objects store metadata about the data in the base tables, describing where pertinent data is located and the relationships within the base data.

DWE Design Studio

DWE Design Studio is an integrated development environment (IDE) for data warehouse projects that integrates consistent and interoperable tools for:

- Connecting and browsing databases
- Exploring data
- Designing physical database models (reverse/forward engineering)
- Designing OLAP objects
- Designing and deploying Data Mining flows

- Designing and deploying SQL Warehousing data and control flows

Eclipse is a universal platform for integrating tools, providing the powerful framework and the common GUI and infrastructure required to integrate tools. This platform can easily be extended by installing plug-ins provided by the platform itself or by other plug-ins.

The framework and general workbench of Eclipse is developed and maintained by a large community of companies, including IBM. As an Eclipse based tool, DWE Design studio is easily extended with third party tools.

DWE Design Studio aggregates tools to handle SQL tasks, Mining Editors, OLAP Tools, and Data Exploration Tools, as depicted in Figure 4-3 on page 123.

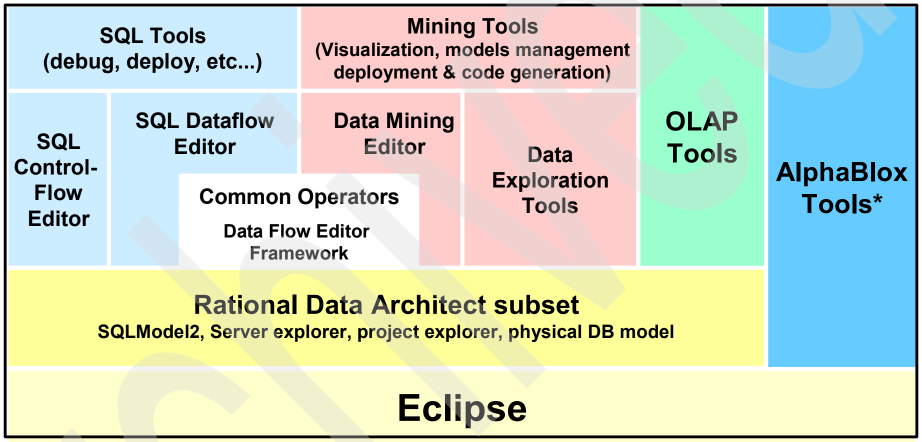


Figure 4-3 DWE Design Studio overview

The DWE Design Studio is a workbench composed of perspectives. The look and feel of this workbench is basically the same as the other tools based on Eclipse, and is depicted in Figure 4-4.

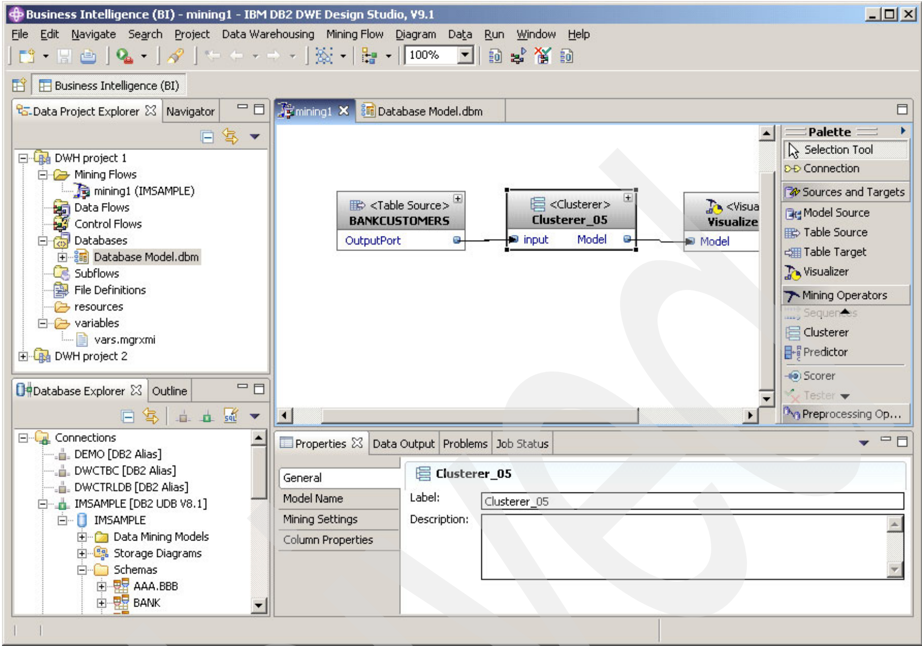


Figure 4-4 DWE Design Studio workbench

DWE Design Studio perspectives are arrangements of views and editors, with different perspectives being suited for different tasks and delivering the right tool for a specific job. Although the deployed standard perspectives are very comprehensive, you can still customize them if required.

Business Intelligence perspective

In this particular BI perspective all DWE related activities are aggregated, as can be seen in Figure 4-5.

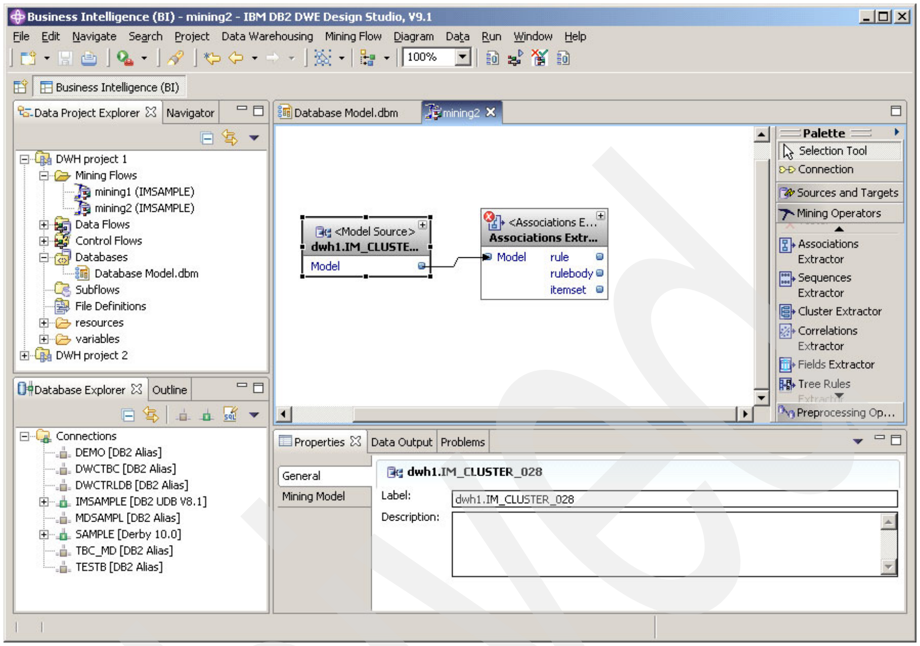


Figure 4-5 Business Intelligence perspective

The outline view shows an overview of the structure of the document which is currently edited. The appearance of the view depends on the nature of the edited document.

When the flow is edited, the outline offers two representations:

1. Graphical, showing the entire flow.
2. Tree, where the objects composing the flow can be easily selected.

These representations are depicted in Figure 4-6.

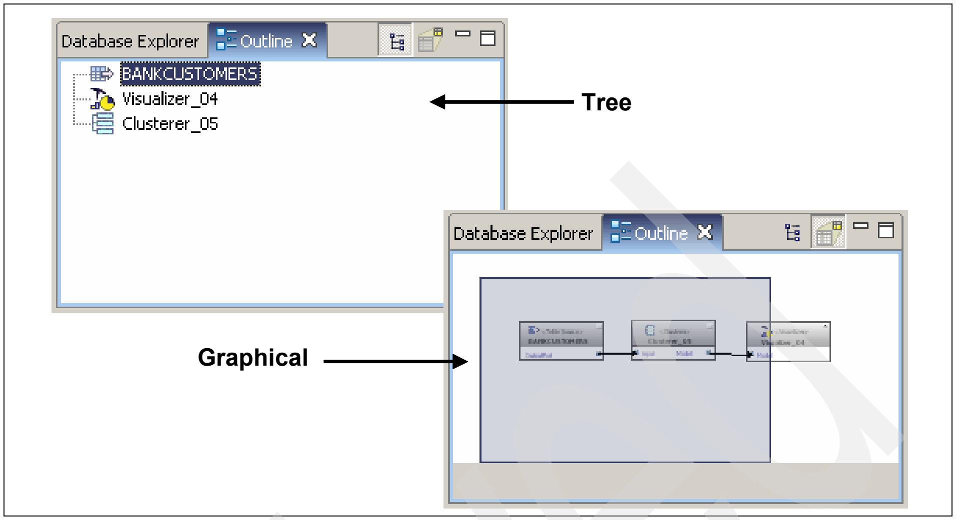


Figure 4-6 Outline view representations: Tree and Graphical

The properties view allows you to view and modify the properties of the current selected object. The appearance of this view is dependent on the current selection. This view is one of the most important views inside the BI perspective when designing flows or database objects, and can be seen in Figure 4-7.

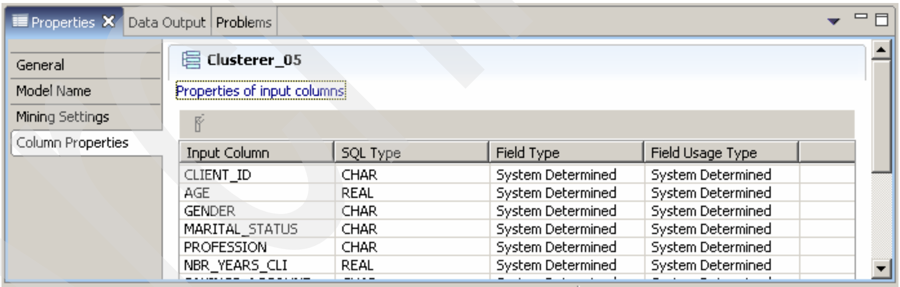


Figure 4-7 Properties view

The Data Output view displays the result of SQL statements when they are executed on a database. This view is used in the following scenarios:

- Inspect the content of a table
- Execute data mining flows
- Execute SQL/DDL scripts or to perform any operation on the database

The history of the last run queries is kept on the left hand side of the view.

The right hand side of the view displays a sampling of an eventual result set returned by the query and, under messages, the full SQL statement that has been executed, along with the eventual messages returned by the database. This view is depicted in Figure 4-8.

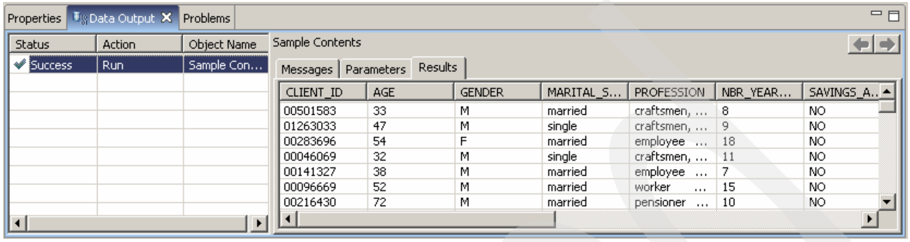


Figure 4-8 Data output view

There is also a problems view. This view shows any errors and warnings detected during validation of the resources contained in the opened projects. There are three levels of severity messages, errors, warnings and information.

You can sort the messages by severity, description or resources. To check the message, click it and the resource containing the problem will open. An example of this view is depicted in Figure 4-9.

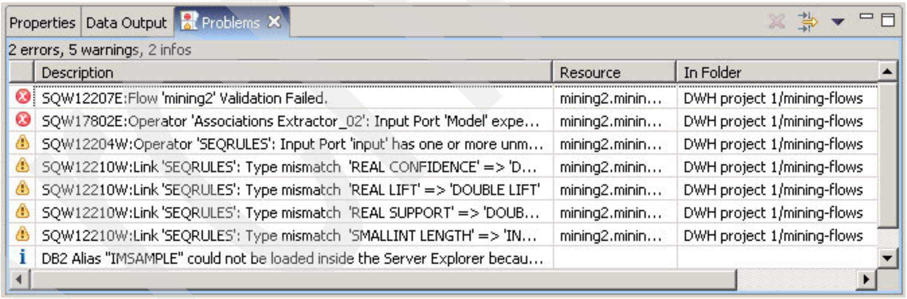


Figure 4-9 Problems view

Data Modeling

Using the features enabled you can create new database designs or use reverse engineering. You can create diagrams from existing schemata, such as UML notations, and modify entities graphically.

Another feature enabled is the ability to compare database objects with objects created on an existing database. A sample project using this feature is depicted in Figure 4-10.

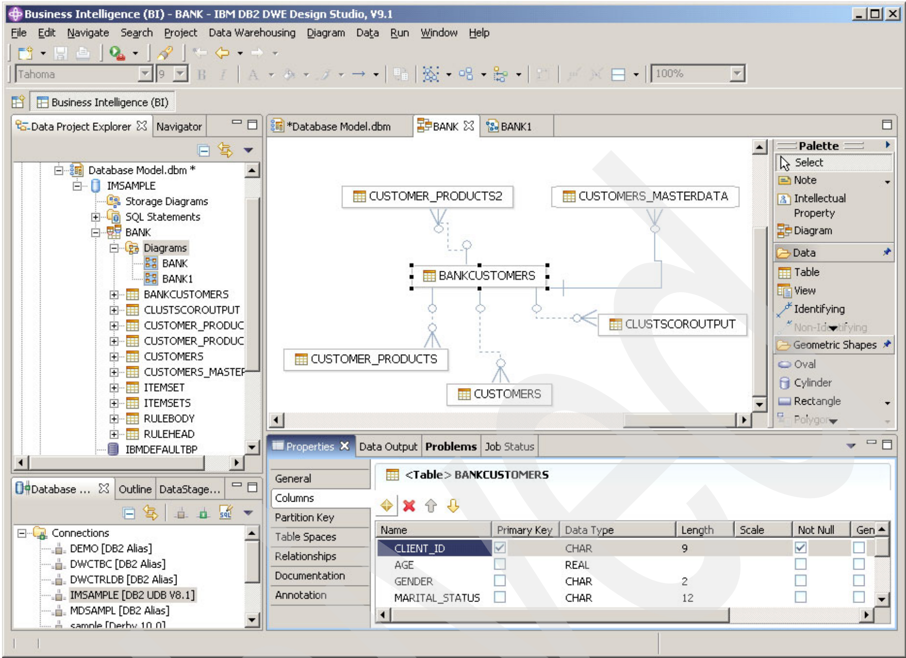


Figure 4-10 Data Modeling perspective

In this perspective you can also manage physical entities, such as tablespaces and containers. This is depicted in Figure 4-11.

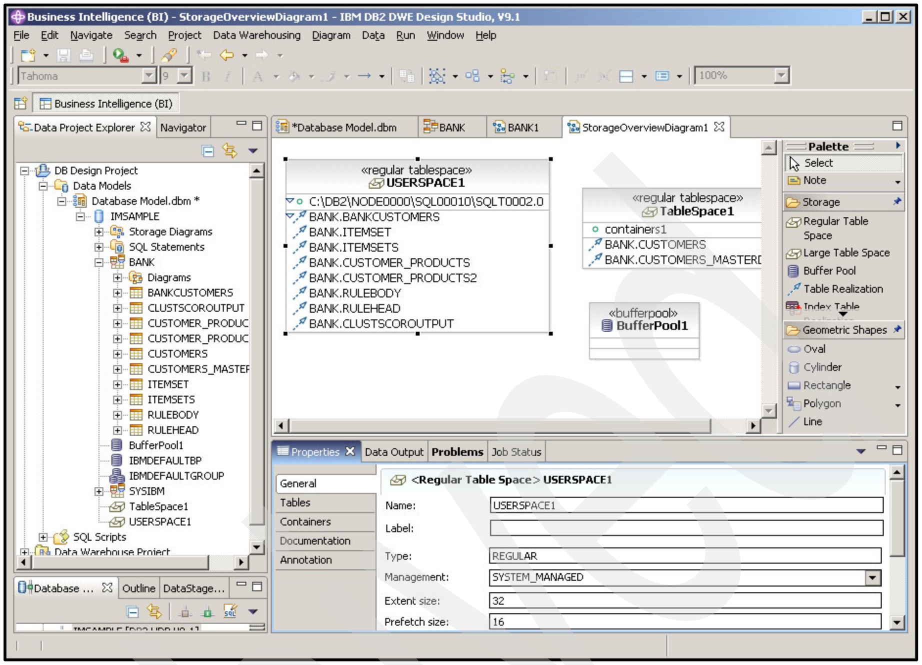


Figure 4-11   Physical modelling

OLAP modeling is an extension of the data modeling capabilities used to design OLAP cubes in Physical Data Models. A sample is depicted in Figure 4-12.

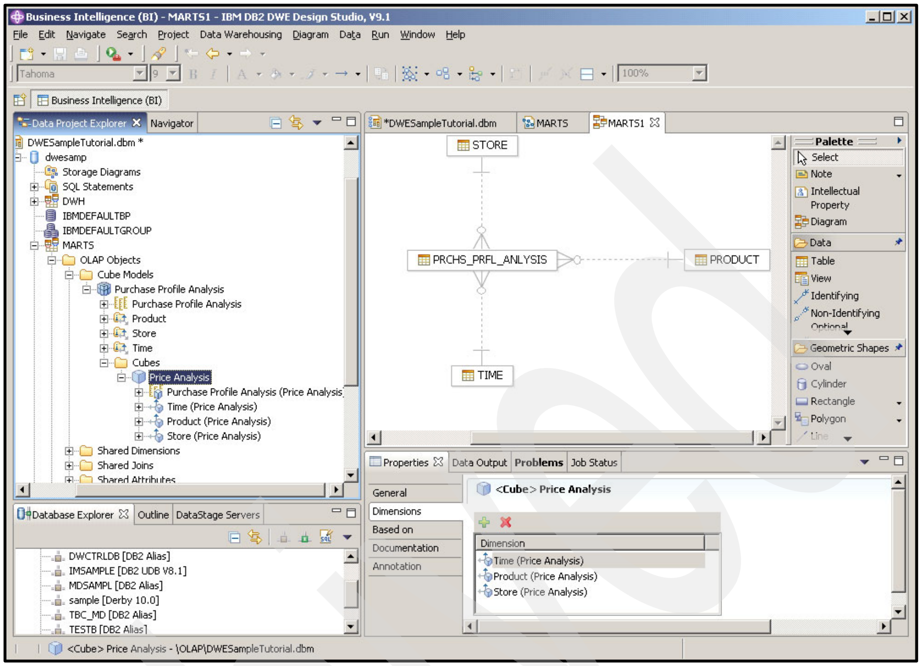


Figure 4-12 OLAP modelling

DWE SQL Warehousing Tool

The SQL Warehousing Tool (SQW) is used to control and generate deployable packages that are typically composed of data flows and control flows. A data flow is used to define data transformation steps through the use of a library of operations typically used for the most common data extraction and transformation steps. For more comprehensive operations, this library can be extended.

You can use general SQL operators to directly express transformations in SQL and also create reusable sub-flows for often used transformation patterns, helping to decrease complexity and maintenance.

A sample data flow is depicted in Figure 4-13.

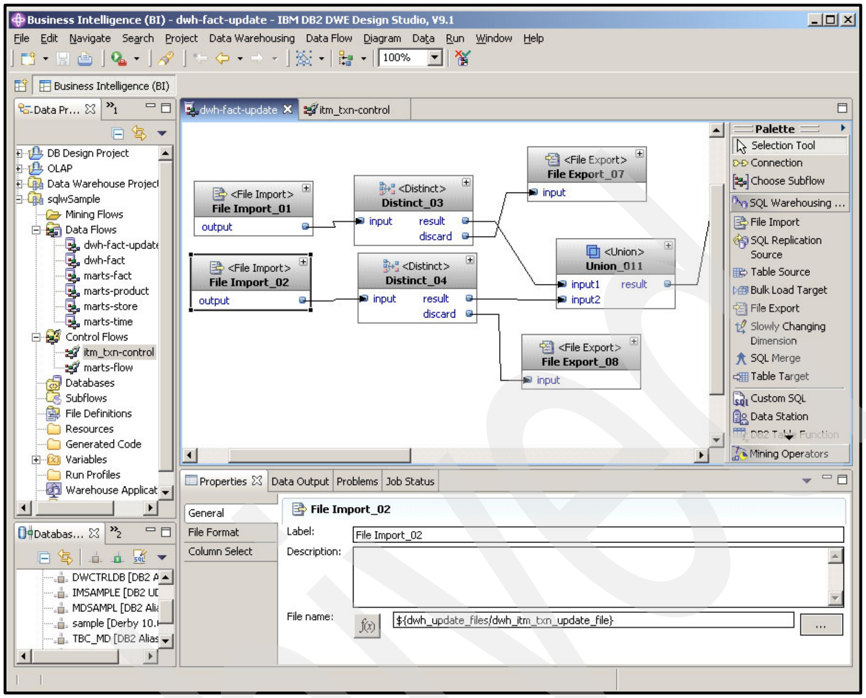


Figure 4-13 Sample data flow

Another type of flow is also generated by DWE Design Studio, and is used to control and coordinate the execution of several data flows.

In this flow you have support for execution conditions, such as *on success*, *on failure*, and *always,* to create logical flows.

A sample control flow is depicted in Figure 4-14.

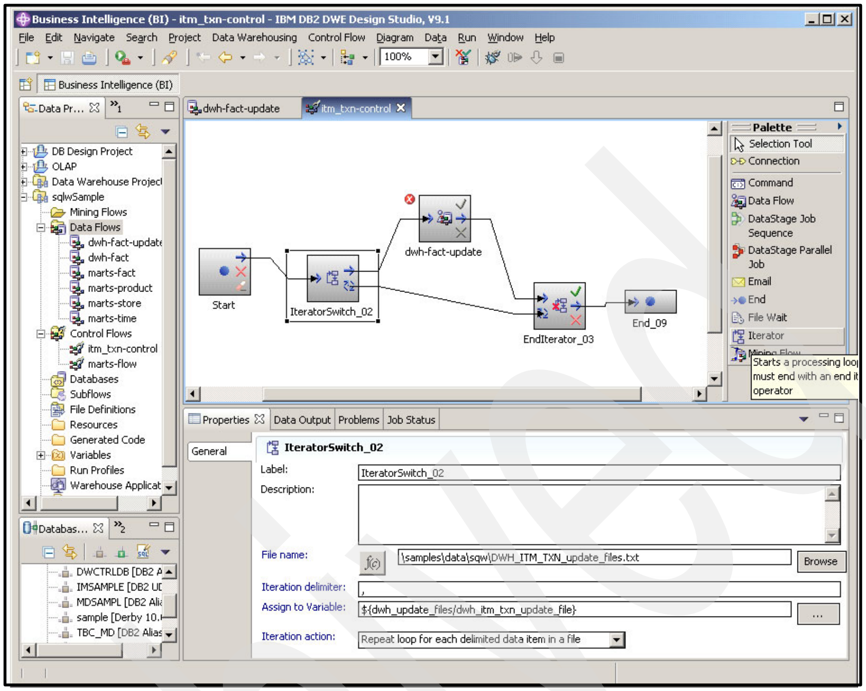


Figure 4-14 Sample control flow

For more information about data warehousing features, see “DWE SQL Warehousing Tool” on page 144.

Projects

Before you can design data warehouse resources you have to create a project. The following is a list of sample project types:

- Data Design: Design database physical models, including OLAP modeling, to execute forward/reverse engineering over database models.
- Data Warehouse: Design SQL Warehouse and data mining flows. This project type is also used to generate Data Warehouse Application packages to deploy on DWE servers.

A project creation sample can be seen in Figure 4-15.

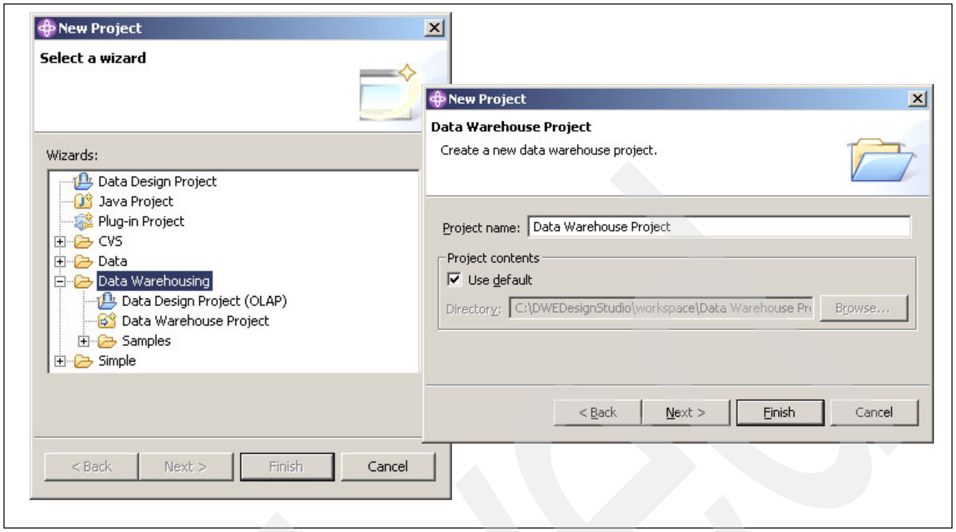


Figure 4-15 Creating a project

DWE Integrated Installer

DWE Integrated Installer is responsible for installing numerous products from a single interface.

The first screen of DWE Integrated Installer is show in Figure 4-16. Note that you can choose between Enterprise Edition and Base Edition.

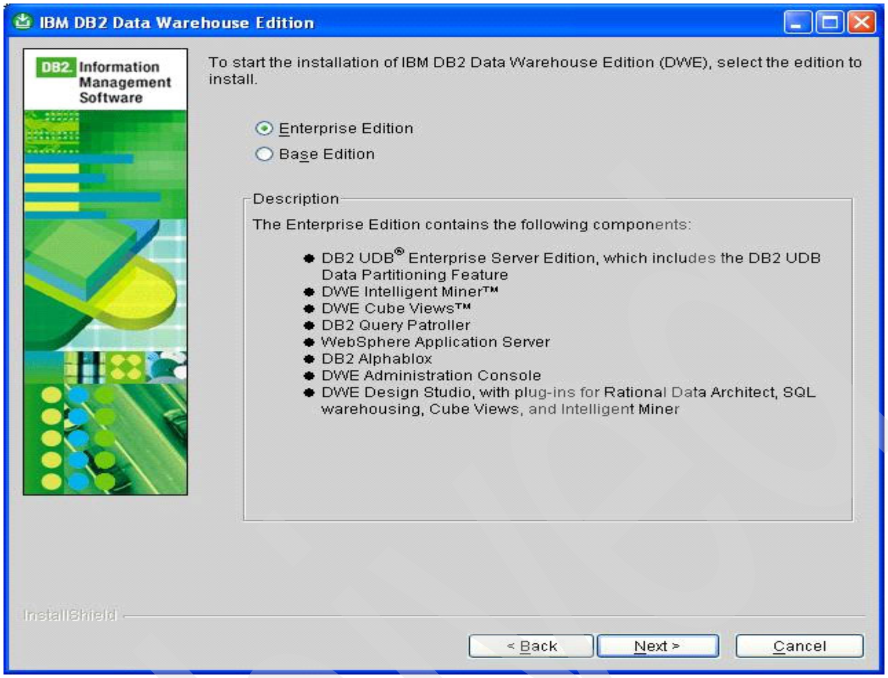


Figure 4-16 Integrated installer

Through the use of this tool, you can select several features that can be used when you have distinct servers for specific roles on the BI scenario. You can also customize the components that you will be using on the working server. The features available are depicted in Figure 4-17.

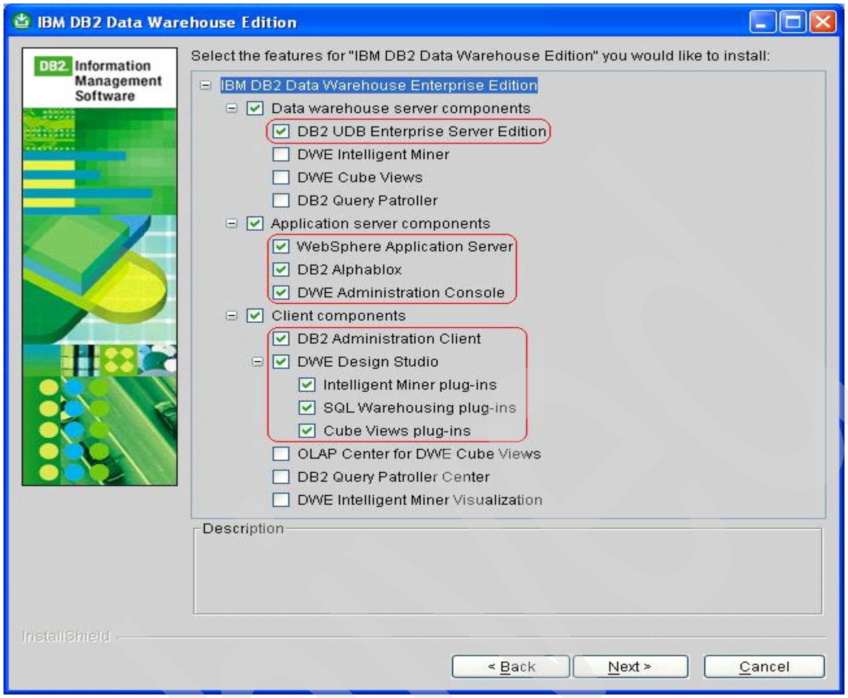


Figure 4-17 Customizing installation features

DB2 Data Partitioning Feature

A database partition is part of a database that consists of its own data, indexes, configuration files, and transaction logs. It is sometimes called a node or a database node.

A partitioned database is a database with two or more partitions. Tables can then be located in one or more of the database partitions. Processors associated with each database partition are used to satisfy table requests. Data retrieval and update requests are decomposed automatically into sub-requests, and executed in parallel among the applicable database partitions. In this type of database, data is hashed for storage.

Tables can be located in one or more database partitions. When table data is distributed across multiple partitions, some of its rows are stored in one partition, and other rows are stored in other partitions. Data retrieval and update requests are decomposed automatically into sub-requests, and executed in parallel among the applicable database partitions. The fact that databases are split across database partitions is transparent to users.

Typically, a single database partition exists on each physical component that makes up the computer. The processors on each system are used by the database manager at each database partition to manage its part of the total data in the database. Because data is divided across database partitions, you can use the power of multiple processors on multiple computers to satisfy requests for information.

Data retrieval and update requests are decomposed automatically into sub-requests and are executed in parallel among the applicable database partitions.

User interaction occurs through one database partition, known as the coordinator partition for that user. The coordinator runs on the same database partition as the application, or, in the case of a remote application, the database partition to which that application is connected. Any database partition can be used as a coordinator partition.

DPF allows great flexibility in spreading data across multiple partitions (nodes) of a partitioned database. Users can choose how to partition their data by declaring partitioning keys, and can determine which and how many partitions their table data can be spread across by selecting the database partition group and tablespace in which the data should be stored. In addition, a partitioning map (which is updatable) specifies the mapping of partitioning key values to partitions. This makes it possible for flexible workload parallelization across a partitioned database for large tables, while allowing smaller tables to be stored on one or a small number of partitions if the application designer so chooses. Each local partition may have local indexes on the data it stores to provide high performance local data access.

A partitioned database supports a partitioned storage model, in which the partitioning key is used to partition table data across a set of database partitions. Index data is also partitioned with its corresponding tables, and stored locally with each partition.

Before partitions can be used to store data, they must be defined to the database manager. Partitions are defined in a file called *db2nodes.cfg*.

The partitioning key for a table in a tablespace on a partitioned database partition group is specified in the CREATE TABLE statement or the ALTER TABLE statement. If not specified, a partitioning key for a table is created by default from the first column of the primary key. If no primary key is defined, the default partitioning key is the first column defined in that table that has a data type other than a long or a LOB data type. Partitioned tables must have at least one column that is neither a long nor a LOB data type. A table in a tablespace that is in a single partition database partition group will have a partitioning key only if it is explicitly specified.

Hash partitioning is used to place a row in a partition as follows:

1. A hashing algorithm (partitioning function) is applied to all of the columns of the partitioning key, which results in the generation of a partitioning map index value.
2. The partition number at that index value in the partitioning map identifies the partition in which the row is to be stored.

DB2 DPF supports partial de-clustering, which means that a table can be partitioned across a subset of partitions in the system (that is, a database partition group). Tables do not have to be partitioned across all of the partitions in the system.

DB2 DPF has the capability of recognizing when data being accessed for a join or a subquery is located at the same partition in the same database partition group. This is known as table collocation. Rows in collocated tables with the same partitioning key values are located on the same partition. DB2 DPF can choose to perform join or subquery processing at the partition in which the data is stored. This can have significant performance advantages.

Collocated tables must:

- Be in the same database partition group, one that is not being redistributed. (During redistribution, tables in the database partition group may be using different partitioning maps - they are not collocated.)
- Have partitioning keys with the same number of columns.
- Have the corresponding columns of the partitioning key be partition compatible.
- Be in a single partition database partition group defined on the same partition.

DB2 Query Patroller

DB2 Query Patroller is a powerful query management system that you can use to proactively and dynamically control the flow of queries against your DB2 database in the following key ways:

- Define separate query classes for queries of different sizes to better share system resources among queries and to prevent smaller queries from getting stuck behind larger ones
- Give queries submitted by certain users higher priority
- Automatically put large queries on hold so that they can be canceled or scheduled to run during off-peak hours
- Track and cancel runaway queries

The features of Query Patroller allow you to regulate the database query workload so that small queries and high-priority queries can run promptly and your system resources are used efficiently. In addition, information about completed queries can be collected and analyzed to determine trends across queries, heavy users, and frequently used tables and indexes.

Administrators can use Query Patroller to:

- Set resource usage policies at the system level and at the user level.
- Actively monitor and manage system usage by canceling or rescheduling queries that could impact database performance.
- Generate reports that assist in identifying trends in database usage, such as which objects are being accessed, and which individuals or groups of users are the biggest contributors to the workload.

Query submitters can use Query Patroller to:

- Monitor the queries they have submitted.
- Store query results for future retrieval and reuse, effectively eliminating the need for repetitive query submission.
- Set a variety of preferences to customize their query submissions, such as whether to receive e-mail notification when a query completes.

Query Patroller components

DB2 Query Patroller is a client and server solution that can be deployed on a system running DB2 Enterprise Server Edition. It consists of the following components:

- Query Patroller server, which consists of:
  - Stored procedures to perform the necessary database tasks.
  - Control tables with the information to manage queries. For example, the SUBMITTER_PROFILE table contains information such as the submitter ID, authority level, and the maximum number of queries that the user can have running simultaneously. When the user submits a query, Query Patroller references the SUBMITTER_PROFILE table for these parameters.
  - Log files to hold error information
- Query Patroller Center, is a graphical user interface that allows administrators to manage Query Patroller system properties, users, and queries, and to view historical analysis reports. It also allows query submitters to manage their queries, save query results, and customize their query submission preferences.

The look and functionality of the Query Patroller Center varies depending on different factors, such as the authority of the user and whether the DB2 administration tools are also installed.

An administrator can perform, as examples, the following tasks:

– Manage the Query Patroller system parameters.
– Create, update, or delete profiles.
– Create, update, or delete submission preferences.
– Create, update, or delete query classes.
– Monitor and manage intercepted queries.
– Generate and analyze reports that display database usage history.

A submitter has access to a subset of the functionality of Query Patroller Center, and can perform, as examples, the following tasks:

– Monitor and manage submitted queries.
– Store results of the queries that they have submitted for future retrieval.
– Show or file results of the queries that they have submitted.
– Create, update, or delete their own query submission preferences.

- Query Patroller command line support, enables administrators and submitters to perform most Query Patroller tasks from the DB2 CLP or from the operating system command line prompt. Query Patroller commands can also be combined with shell scripts or languages such as Perl, awk, and REXX.

DWE Mining

Data mining is embedded into DB2 through the use of DB2 stored procedures and user-defined functions (UDFs). That are three distinct modules:

- Modeling
- Scoring
- Visualization

A typical data modeling process is based on the steps depicted in Table 4-2. As a prerequisite, data should have been pre-processed.

Table 4-2 Common mining steps

step	module	description
1	Modeling	defining a mining task
2	Modeling	doing a mining run and building a model
3	Visualization	visualizing the model
4	Scoring	scoring new data against the model (prediction)

The data mining run is accomplished by:

- DWE Mining Editor
- Easy Mining Procedures
- SQL Stored Procedures

Through the use of the DWE Mining Editor you can compose mining tasks as an integrated part of end-to-end data flows using a graphical canvas. The Easy Mining interface is a high level API composed of Java UDFs with a simplified interface to do a mining run with a single call to a SQL stored procedure.

SQL Stored procedures with SQL/MM API is a detailed expert API comprised of UDFs for the mining tasks, models and test results. UDMs are used for defining data mining tasks and stored procedures to build and test models. UDFs, and table UDFs, are used for analyzing built models such as model signature, model statistics, and model quality. An example of the use of these features is depicted in Figure 4-18.

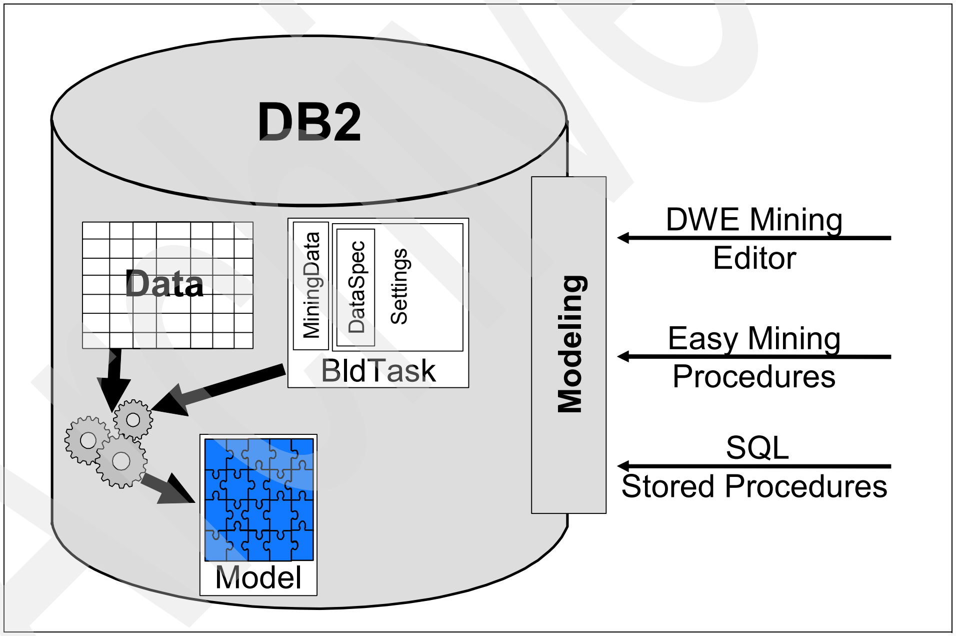


Figure 4-18 Data mining run

The visualization functionality uses Java visualizer for Predictive Model Markup Language (PMML) for full functionality with models created by DWE V9.1 and IBM DB2 Intelligent Miner V8. PMML models from other providers can be visualized. Currently the PMML model versions supported are V2.1, V3.0 and V3.1.

You can read and modify models stored as files in PMML format and in DB2 as objects of the SQL/MM types. Visualization is available as standalone application as part of DWE V9.1 or a Java Applet on a HTTP Server. A Visualization sample can be seen in Figure 4-19.

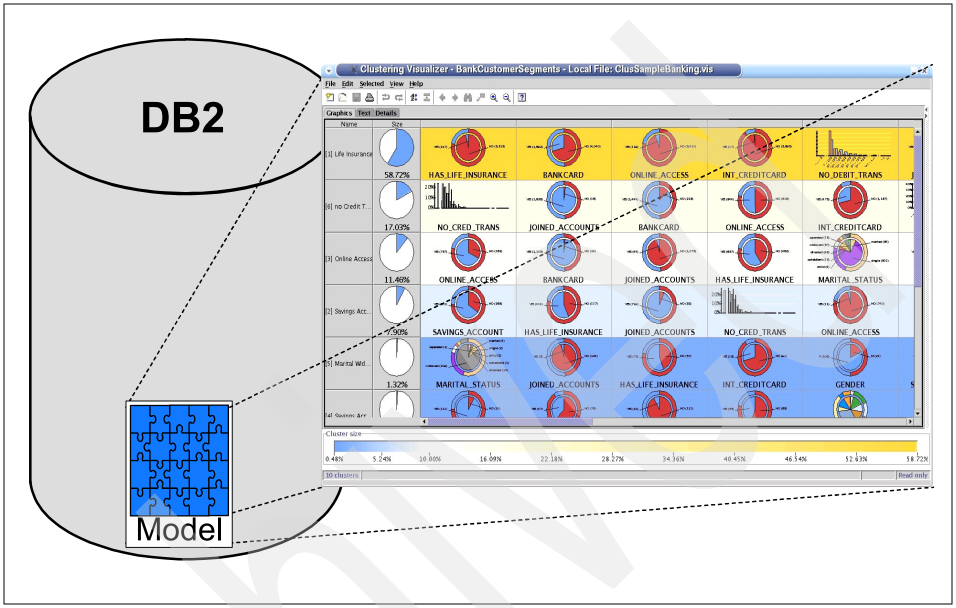


Figure 4-19 Visualizing a model

The Scoring functionality is accomplished by the use of the Easy Mining Interface, which is based on Java UDFs to do a scoring run with a single call to a SQL stored procedure and store the scoring result in a view.

Scoring can also be accomplished with SQL/MM APIs by the use of UDTs for the models, result specs and scoring results, UDFs to import models into DB2, score data against models and analyze the scoring result.

An example of scoring is depicted in Figure 4-20.

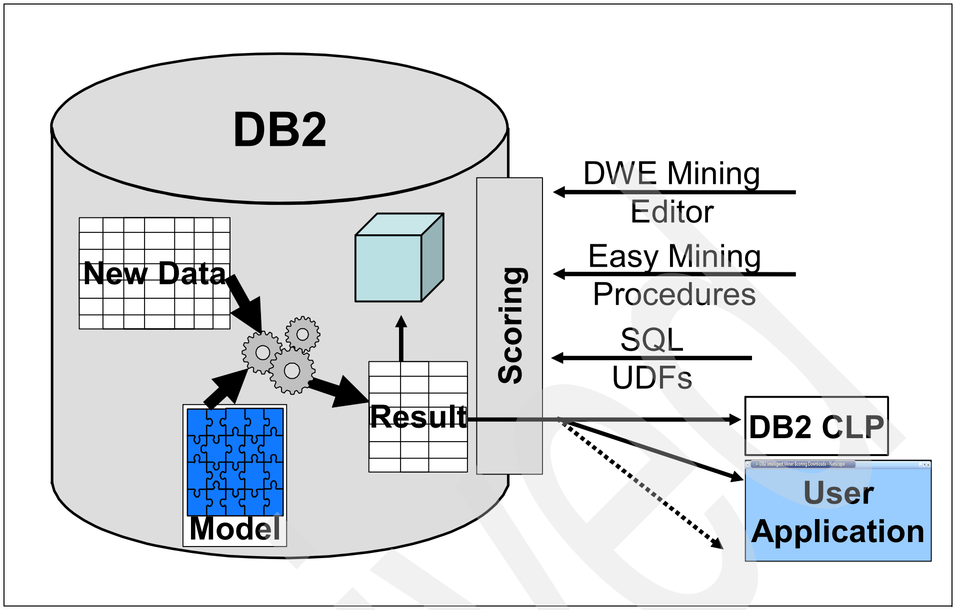


Figure 4-20 Scoring new data against a model

DB2 Admin Console

The DB2 Admin Console is a J2EE application that runs on an application server. It is developed in Java, and based on Java Server Faces (JSF).

The Admin console is a common interface for administrative tasks which provides a unified one-stop-shopping point of entry and enables remote access via a Web-base client. Its positioning in the DWE structure is depicted in Figure 4-21.

Some features that can be executed on the Admin Console are:

- Management and monitoring SQL Warehousing applications
- Management and optimization OLAP Cube Views models
- Management and Scoring Intelligent Mining

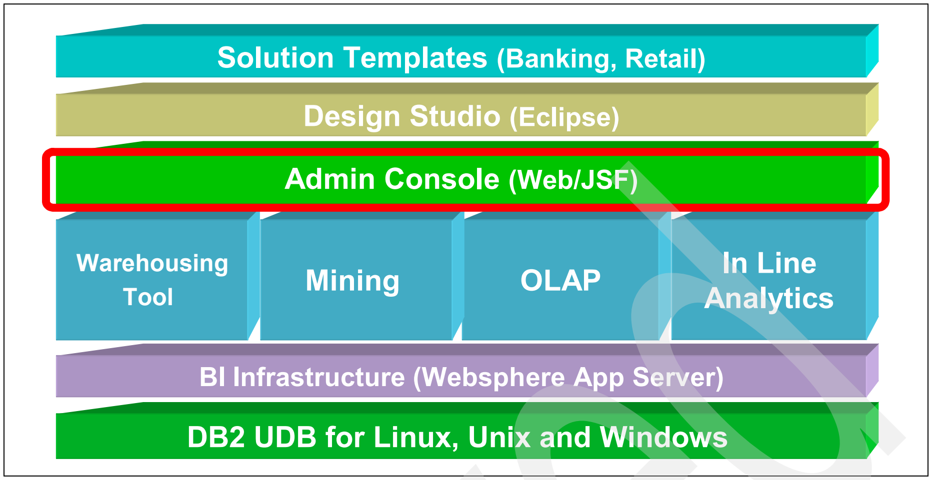


Figure 4-21 Admin Console positioning

As a J2EE application hosted on an application server, you can reach role based security resource management and manage Enterprise Java Beans. The Admin Console components are depicted in Figure 4-22.

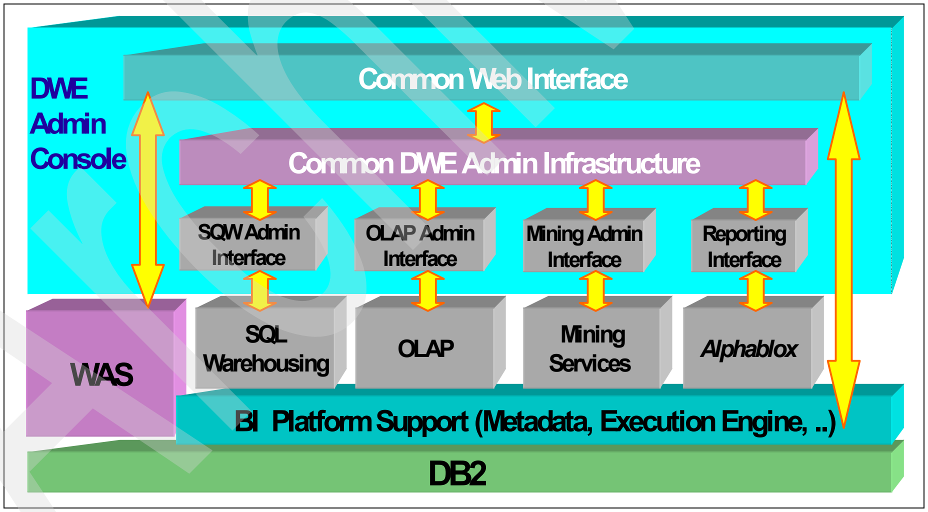


Figure 4-22 DWE Admin Console components

The DWE Admin console resides on the same system as the WebSphere application server and is deployed to WebSphere application server as an enterprise application which leverages WebSphere technology to access resources such as databases, scheduler, work manager and mail provider.

DWE Data Integration Service (DIS) can access the data sources via WebSphere application interface or DB2 application interface.

As depicted in Figure 4-23,the execution/source/target databases referenced by data flow activities can be local or remote to the DWE Admin Console and connections to these databases can be managed by WebSphere or DIS.

The target databases for SQL script activities can also be local or remote to the DWE Admin Console. The connections to these target databases are managed by DIS, and the control database for DWE can reside locally or remotely.

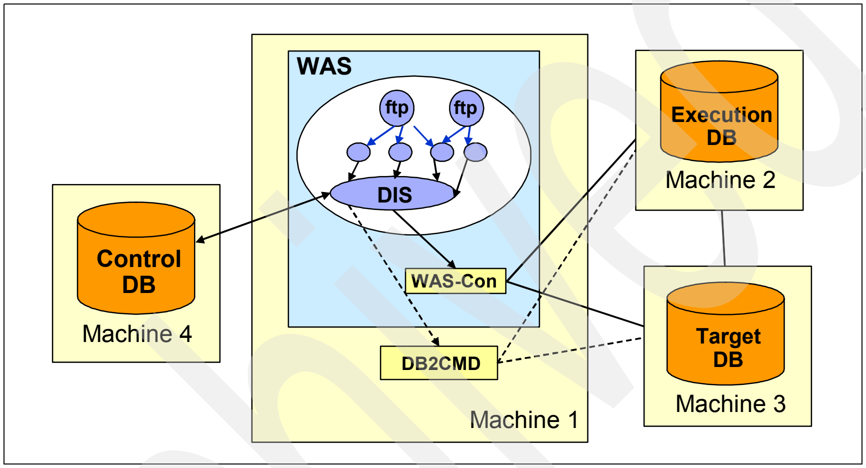


Figure 4-23 DWE Admin Console topology

DWE SQL Warehousing Tool

DWE SQL Warehousing Tool (SQW) can be used to build and maintain data warehouses in DB2 contexts, as well components and configuration.

Some features and functions of SQW are:

- Interaction with physical modeling through the use of DWE Design Studio, OLAP and mining functions in an integrated Eclipse-based GUI.
- Code and execution plan generation optimized for DB2.
- In-database (DB2) SQL transformations and updates for data in warehouse and mart tables.
- WebSphere Application Server runtime environment for role-based application management (deployment, scheduling, monitoring) via Administration Console.
- Integration points with WebSphere DataStage® ETL engine.

Components needed to created SQW are:

- DWE Design Studio
- DWE Administration Console
- DB2 UDB
- WebSphere Application Server

The life cycle of a data warehouse application using SQW can be segmented as shown in Table 4-3.

Table 4-3 Lifecycle of a data warehouse application steps

Step	Design Studio	Admin Console	DB2 Instance	WAS	Description
1	Needed	Needed	Needed	Needed	Install design and runtime environments
2	Needed	Not Needed	Needed	Not Needed	Design and validate data flows
3	Needed	Not Needed	Needed	Not Needed	Test-run data flows
4	Needed	Not Needed	Needed	Not Needed	Design and validate control flows
5	Needed	Not Needed	Needed	Not Needed	Test-run control flows
6	Needed	Not Needed	Not Needed	Not Needed	Prepare control flow application for deployment
7	Not Needed	Needed	Needed	Needed	Deploy application
8	Not Needed	Needed	Needed	Needed	Run and manage application at process (control flow) level
9	Needed	Needed	Needed	Needed	Iterate based on changes in source and target databases

Data Flows

Data flows are models that translate data movement and transformation requirements into repeatable, SQL-based warehouse building processes. Data from source files and tables moves through a series of transformation steps before loading or updating a target file or table.

The example depicted in Figure 4-24 selects data from a DB2 staging table, removes duplicates, sorts the rows and inserts the result into another DB2 table. Discarded duplicates go to a flat file.

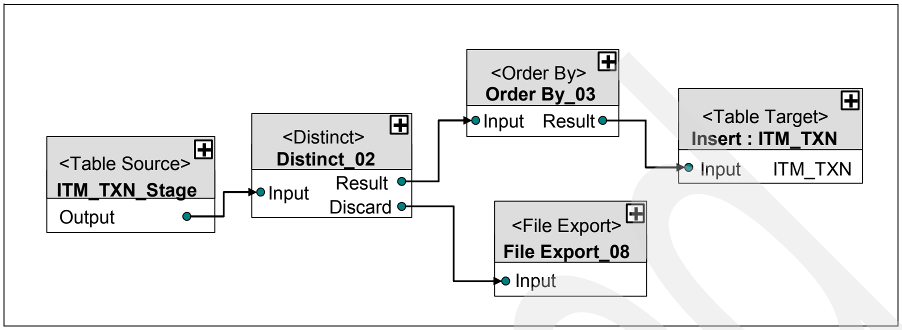


Figure 4-24   simple data transformation example

The possible sources and targets for Data Flows are:

- Sources
  - File import
  - Table Source
  - SQL replication source
- Targets
  - File export
  - Table target (insert, update)
  - Bulk load target (DB2 load utility)
  - SQL merge
  - Slowly changing dimension (SCD)
  - Data station (special staging operator, intermediate target)

To execute data flows in DWE Design Studio you must:

- have no errors listed on the problem view
- choose or define the run profile
- select resources and variable values, if required

The execution in DWE Design Studio is intended for testing purposes. To promote them to production you must deploy them through the use of the DWE Administration Console hosted on a WebSphere Application Server.

Control flows

A control flow is a container model that sequences one or more data flows and integrates other data processing rules and activities.

In Figure 4-25 there is a simple example that processes two data flows in sequence. If they fail, an email is sent to an administrator.

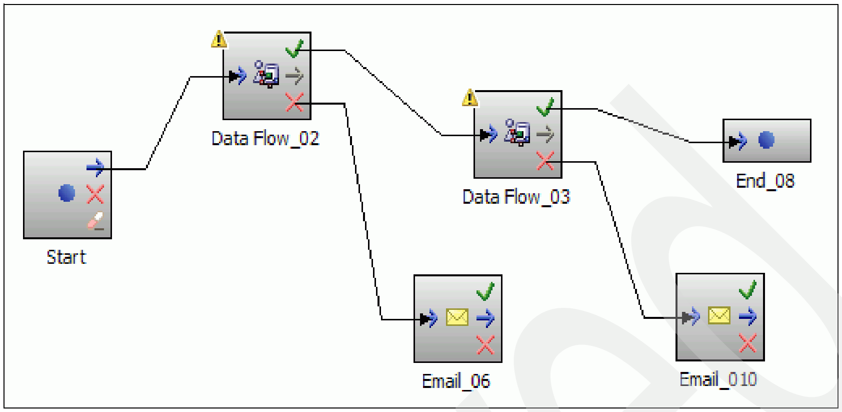


Figure 4-25 Simple control flow example

Control flows has several standard operators, such as:

- Start and end placeholders
- Mining flow
- Command (shell scripts, executables)
- Data Flow
- DataStage job sequence
- Email
- DataStage parallel job
- Iterator
- File wait

The operations palette is depicted in Figure 4-26:

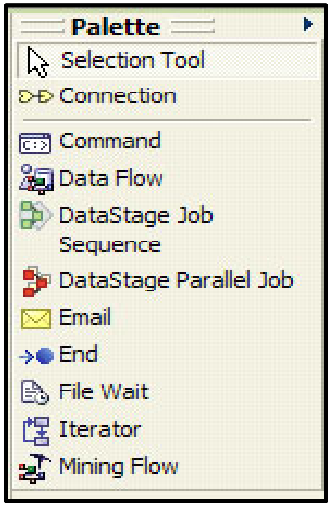


Figure 4-26 Control flow operators

Deployment

Deployment of SQW applications are executed based on the following steps:

1. Define required data sources and system resources in the WebSphere Application Server environment.
2. Enter the location of the zip file that was created by the preparation step in the Design Studio.
3. Link the data sources that the application needs to the data sources previously created.
4. Link the system resources (if any) that the application needs to the system resources previously created.
5. If your application contains variables that can be altered after the design and deployment preparation phases, you need to enter variable values or accept the current default values.

4.2 DB2 Alphablox

DB2 Alphablox and all Alphablox analytic-enabled solutions run as J2EE-compliant applications in an application server, and they are accessed by using a Web browser. Unlike traditional query and reporting tools that interact with application servers, DB2 Alphablox leverages the application services, portal services and integration broker services provided by the application server. In addition, DB2 Alphablox leverages the common foundation for developing, deploying and maintaining distributed applications.

DB2 Alphablox architecture

DB2 Alphablox is comprised of the following elements:

- Platform
- Analytic-enabled solutions
- Administration application
- Application server adapters

The platform, the core component of DB2 Alphablox, runs within the business tier of the J2EE application server. While running as a J2EE application within the host application server, it also provides the services of a fully functional analysis server. For DB2 Alphablox analytic-enabled solutions to fully leverage the analytic capabilities and services of DB2 Alphablox, the platform requires a separate installation for components and adapters that are not traditionally part of J2EE applications. These components are depicted in Figure 4-27.

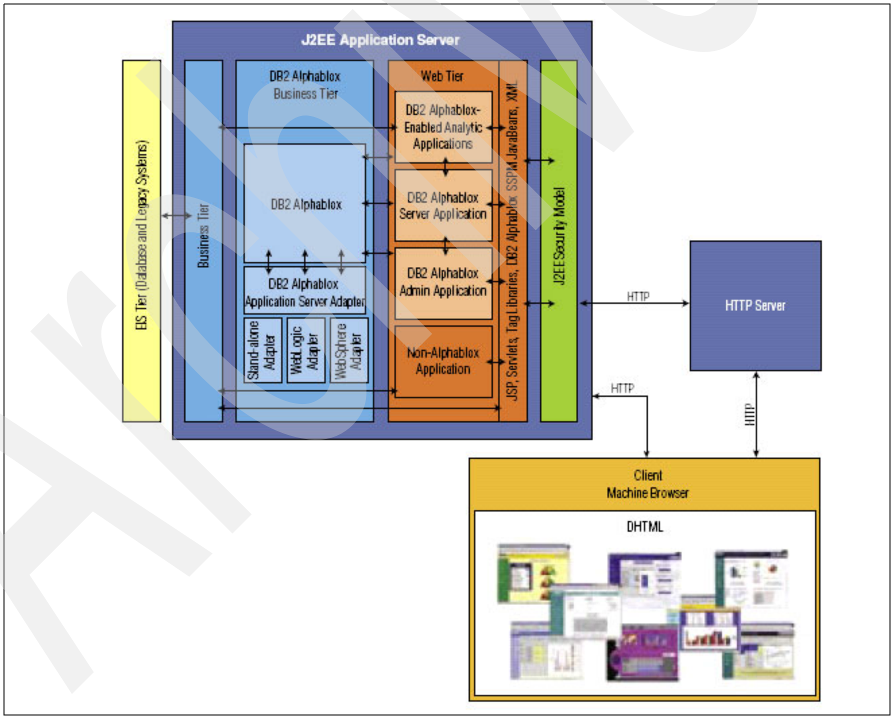


Figure 4-27 DB2 Alphablox architecture

The adapters allow DB2 Alphablox to communicate with each supported application server to perform administration functions. Many of these functions, such as defining applications, are set up differently on each application server.

DB2 Alphablox analytic-enabled applications run as application server applications within the Web tier. The applications, while interacting with DB2 Alphablox, are configured as separate and encapsulated J2EE applications. Updates to DB2 Alphablox-enabled applications can be deployed, backed up, upgraded and migrated independently of the DB2 Alphablox platform.

DB2 Alphablox also registers two J2EE applications within the Web tier of the application server. They are the DB2 Alphablox server application and the DB2 Alphablox administration application. The application server manages DB2 Alphablox in the same way it manages any other Web application. For example, it is auto-started by invoking a servlet. DB2 Alphablox is then suspended and resumed by the application server as needed, based on requests received by the application server and the management model.

DB2 Alphablox analytic components

DB2 Alphablox enables organizations to integrate analytics across functions and lines of business, and deploy analytic solutions for improved decision making. The technology enables organizations to optimize various aspects of their business, including:

- Self-service reporting and analysis applications
- Operational analysis applications
- Financial reporting and analysis applications
- Planning applications
- Business performance and key performance indicators (KPI) for interactive information dashboards

Data can be presented in several formats, including:

- Interactive grids, charts and reports
- Informational dashboards
- Planning and modeling applications
- Information portals

DB2 Alphablox can integrate data from all enterprise information resources, including relational and multidimensional databases, transaction systems and other outside content feeds. This ensures that users have immediate access to all pertinent data, regardless of where or how it is stored. In addition, users can utilize a write-back capability to facilitate real-time planning and modeling applications.

4.2.1 DB2 Alphablox applications

DB2 Alphablox-enabled applications typically have the following characteristics, that may be implemented using various combinations of DB2 Alphablox features:

- Interactive and guided analysis
- Real-time data access, analysis and alerts
- Personalization
- Sharing and collaboration
- Real-time planning through write-back

Interactive and guided analysis

DB2 Alphablox-enabled applications enable users to interact with real-time data via grids and charts, as well as other components such as drop-down lists.

These interactive analytic components are served in dynamic HTML, based on Dynamic HTML (DHTML) technology, utilizing JavaScript™ and cascading style sheets (CSS). The DB2 Alphablox Dynamic HTML client provides the benefits of easy deployment with interaction.

For example, a user can interact with a grid and have just that grid updated rather than having to refresh the entire page.

Users perform multidimensional analysis by manipulating the data displayed in the grid and chart, as depicted in Figure 4-28. Analysis actions such as drilling, pivoting, sorting and selecting can be performed directly on the grid and chart, through toolbar buttons, through right-click menu options or via the DB2 Alphablox form-based controls and components added by application developers.

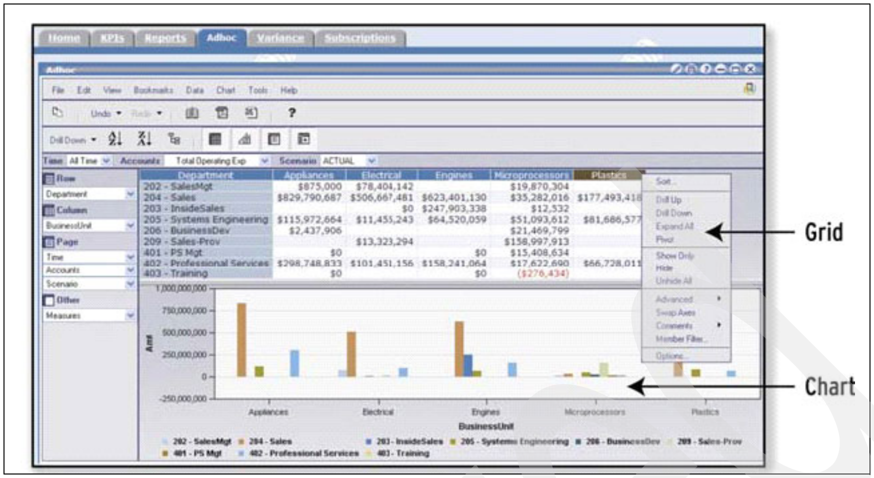


Figure 4-28 DB2 Alphablox Grid and Chart

Real-time data access and analysis

DB2 Alphablox-enabled applications can drive analysis of data from multiple data sources, both relational and multidimensional, including DB2 Cube View cubes. Through the native ability to query a database, DB2 Alphablox exposes the analytic functionality in the database engine. Users can leverage capabilities such as ranking, derived calculations, ordering, filtering, percentiles, variances, standard deviations, correlations, trending, statistical functions and other sophisticated calculations while performing analyses.

For example, a controller of a manufacturing company could choose to look at key performance indicators (KPIs) such as profit, bookings, billings, backlogs, trends and comparisons of actual to budget, as depicted in Figure 4-29. The data is real-time and the controller can choose to drill down on various items, such as total revenue, to get more detail.

Figure 4-29   DB2 Alphablox example: comparisons of actual to budget

The DHTML client in DB2 Alphablox is very flexible. Data can be presented the way users need to see it. For example, in Figure 4-30 a controller wanted to see a butterfly report in which the prior three months of actual figures are shown to the left of the Accounts row headers and the current month details are shown to the right of those headers.

Feb ACTUAL	Mar ACTUAL	Apr ACTUAL	Accounts	May ACTUAL	May BUDGET	May Variance	Variai
275,522 K	202,728 K	195,090 K	Payroll & Benefits	230,320 K	298,227 K	67,907 K	
324 K	1,125 K	503 K	Employee Devel	862 K	3,237 K	2,375 K	
73 K	0 K	2 K	Recruiting		2,314 K	2,314 K	
491 K	195 K	197 K	Admin Fees	102 K	265 K	163 K	
23,284 K	11,737 K	(4,285 K)	Professional Fees	6,185 K	6,227 K	42 K	
33,492 K	25,001 K	20,038 K	T & E	21,143 K	20,022 K	(1,121 K)	
6,290 K	8,242 K	10,954 K	Marketing	5,428 K	9,921 K	4,493 K	
539 K	196 K	665 K	Events	187 K	971 K	783 K	
5,176 K	1,637 K	1,186 K	Equipment Cost	1,644 K	4,053 K	2,409 K	
15,109 K	14,974 K	16,754 K	Depreciation Exp	14,009 K	16,913 K	2,904 K	
22,311 K	71,743 K	39,362 K	Occupancy Exp	28,629 K	40,886 K	12,258 K	
10,898 K	7,250 K	7,120 K	Office Exp	8,089 K	12,667 K	4,577 K	
(33,506 K)	(55,078 K)	(48,367 K)	Allocations	(37,837 K)	(57,547 K)	(19,710 K)	
31 K	83 K		Other				

Figure 4-30   DB2 Alphablox butterfly report

Personalization

Users have different data and business needs. Therefore, DB2 Alphablox analytic-enabled solutions can be personalized to meet the needs of each individual user. For example, the first log-on screen that users see can be customized according to their role in the organization. Users in the sales department may see the top five best-selling products or the most profitable regions for the month-to-date. Users in finance may be more interested in monthly summary figures for sales, cost of goods, marketing, payroll and profit, as shown in Figure 4-31.

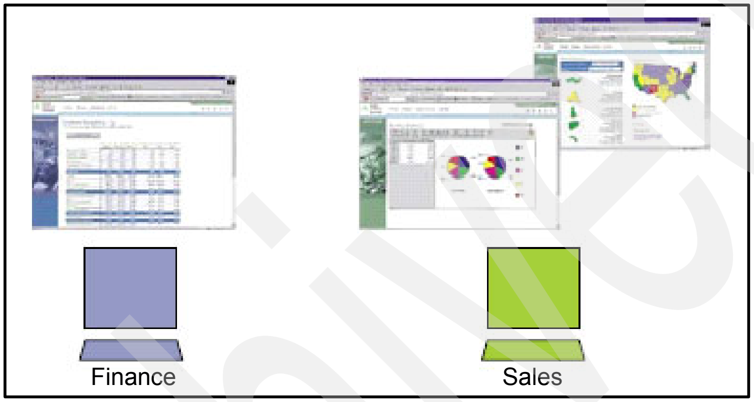


Figure 4-31 DB2 Alphablox customization example

In addition, each DB2 Alphablox analytic-enabled solution may contain custom user preference screens that enable users to personalize the solution to their needs, as depicted in Figure 4-32. In this example, the user can choose the business units and accounts that are displayed in the dial gauges.

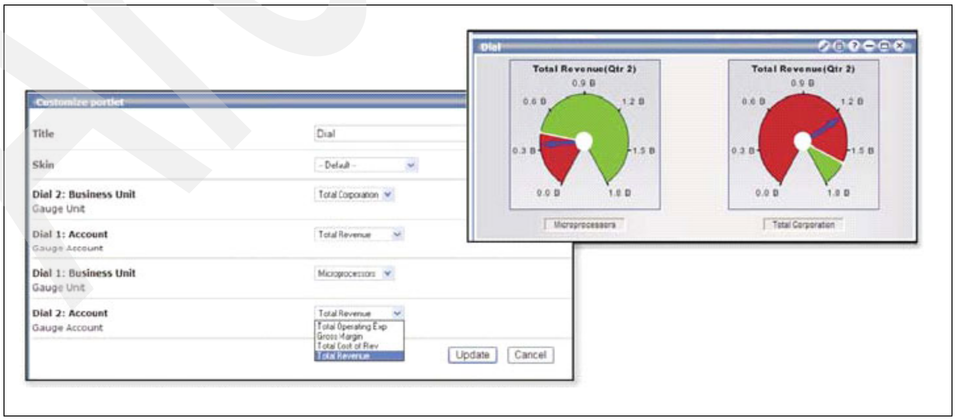


Figure 4-32 DB2 Alphablox personalization example

Sharing and collaboration

DB2 Alphablox analytic-enabled solutions support collaboration, enabling users to leverage existing messaging and workflow systems to save and share application views once the analysis is performed, as depicted in Figure 4-33. In addition, DB2 Alphablox supports collaboration features such as bookmarking, e-mail and PDF generation.

Real-time planning through write-back

Analytic applications may range from historical analysis to forward-looking forecasting and proactive resource allocation. The DB2 Alphablox data write-back capability enables developers to build real-time planning applications such as budgeting, sales forecasting, “what-if” modeling and collaborative demand planning, as seen in Figure 4-33.

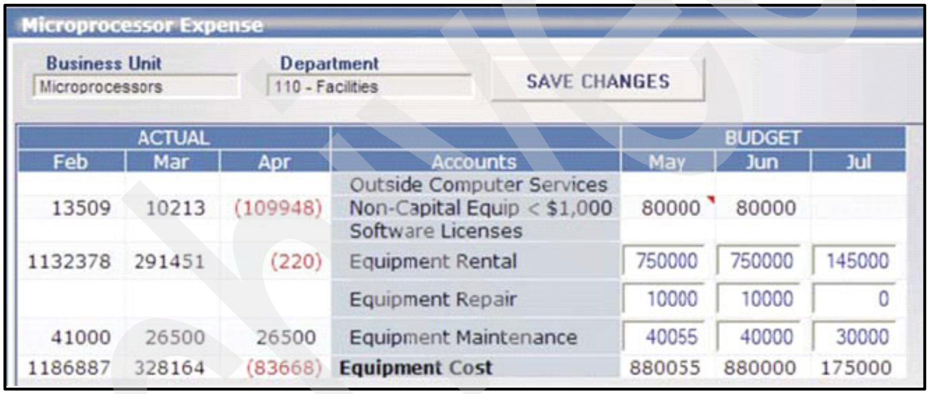


Figure 4-33 DB2 Alphablox what-if modeling example

DB2 Alphablox and the application server

Enterprises gain competitive advantage by quickly developing and deploying custom applications that provide unique business value.

The J2EE standard provides an opportunity for analytic solutions to undergo a true paradigm shift. Prior to J2EE, there was not a standard, cross-platform architecture that would enable truly distributed computing in a Web environment. J2EE simplifies enterprise application development and deployment in several ways:

- Development environment based on standardized, modular components
- A complete set of services to application components
- The ability to extend existing services and add new services that provide complete interoperability with standard services

- Handle the details of application behavior automatically without complex programming

The DB2 Alphablox architecture capitalizes on this standard, cross-platform environment to deliver analytic solutions. DB2 Alphablox draws on Java technologies to implement a Web-based, N-tier architecture for delivery of analytic solutions. J2EE provides the framework for distributed, multi-tiered applications. Application logic is divided into components according to function. The most common configuration is a three-tier configuration as depicted in Figure 4-34, which consists of the following:

1. The Enterprise Information Systems (EIS) tier, also known as the Database tier, runs on database servers. Data resides on these servers and is retrieved from relational and/or multidimensional data servers.
2. The J2EE application server is host to the business and the Web tiers. The business tier is the code that implements the functionality of an application, and the Web tier supports client services through Web containers.
3. The Client tier is where client applications are presented. For a Web-based J2EE application such as DB2 Alphablox, the Web page display and user interface presentation occurs on a client machine through a Web browser. The Web browser downloads Web pages and applets to the client machine.

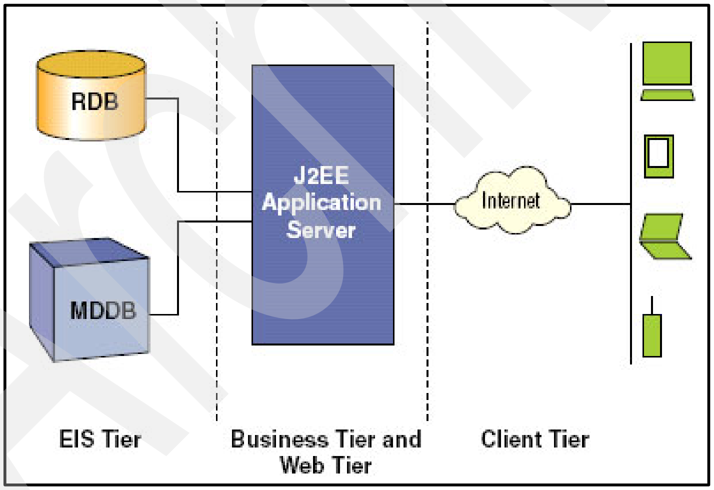


Figure 4-34 Three-tier configuration

Within the J2EE framework, DB2 Alphablox runs as an application within the application server, as shown in Figure 4-35, leveraging existing server resources such as process management, application management and request management. DB2 Alphablox-enabled applications run as standard application server applications and the Web pages are served directly through the application server.

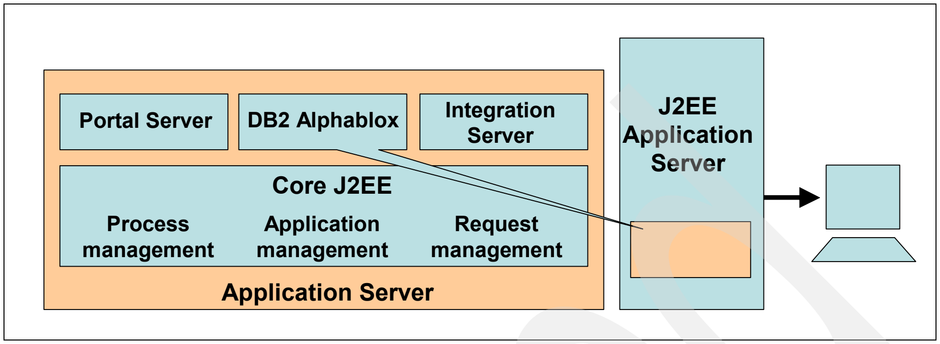


Figure 4-35 DB2 Alphablox running on an application server

Components of a DB2 Alphablox-enabled application

Once installed, DB2 Alphablox provides a comprehensive set of components and application templates for developing analytic solutions. The modular *building Blox* approach enables fast delivery of personalized and customized applications.

DB2 Alphablox analytic-enabled applications appear as a collection of Web pages that serve as containers for the following application components:

- Standard HTML tags and page elements (logos, text or images) to enhance the user interface
- Blox necessary to deliver the required application functionality
- JavaScript for extended application and user interface (UI) logic (optional)
- Java servlets for customized business logic (optional)

Application building Blox

To promote the creation of custom analytic solutions, DB2 Alphablox includes a set of generic application building Blox, as depicted in Figure 4-36. Application building Blox are prebuilt, high-level JavaBean components that provide the functionality required by analytical applications. Blox allow developers to perform data manipulation and presentation tasks and build dynamic, personalized analytic applications. Because Blox are modular and reusable in design, they are easily built into a variety of analytic solutions.

Each Blox provides broad functionality through its properties and associated methods, which allow the Blox appearance and behavior to be specified and controlled. Event filters and event listeners are available for performing pre- and post-event processing for user events such as drilling up or drilling down, pivoting, changing the page filter, loading a bookmark or changing the data value in a grid cell.

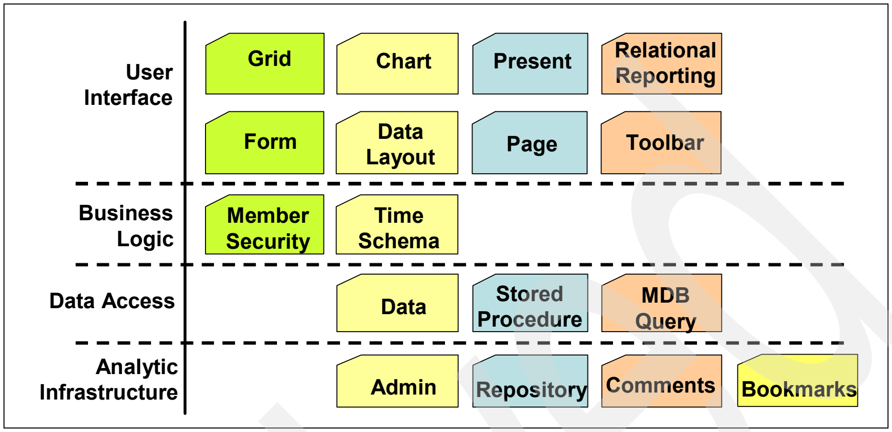


Figure 4-36 DB2 Alphablox application building blox

4.2.2 Deploying DB2 Alphablox

Detailed descriptions of the aspects for deploying DB2 Alphablox-enabled applications are described in the following sections:

Administration

DB2 Alphablox is designed for integration with an existing application server environment to help leverage a robust architecture to deliver analytic solutions. To facilitate installation, DB2 Alphablox provides a cross-platform, graphical interface to install the DB2 Alphablox server and administration pages. Once installed, you have centralized administration of analytic solutions through a set of administration pages. These pages enable application developers to manage DB2 Alphablox services that use the same resources and are also complementary to the administration provided through the application server.

For example, application developers use the DB2 Alphablox administration pages as a convenient way to register and set up new applications. When creating an application from the DB2 Alphablox home page, DB2 Alphablox creates the application definition in the DB2 Alphablox repository. In addition, the J2EE context and directory structure are created.

When running DB2 Alphablox analytic-enabled solutions, the application server passes the user name and associated roles to DB2 Alphablox. To allow personalization, the user profile can be configured to allow the application developer to define custom properties for the user.

DB2 Alphablox administration pages also can be used to configure DB2 Alphablox specific settings such as data sources, relational cubes, groups and DB2 Alphablox server settings. These administration pages, packaged as a J2EE-compliant application, can be managed through the same mechanisms as any other Web application in the environment.

DB2 Alphablox can be administered either through Web pages under the DB2 Alphablox home page or through a standard command console accessible through any telnet terminal software. Using either method, administrators can create users, data sources and other DB2 Alphablox objects. This design enables remote server administration. For example, an application developer can define a new application from a workstation using the Web pages, or an administrator can perform routine monitoring tasks on a remote computer.

Setting up the application

Once the DB2 Alphablox-enabled application is completed, it is a self-contained J2EE Web application that authorized users can access as they would any other Web page. The application developer defines the DB2 Alphablox–enabled analytic application through the appropriate DB2 Alphablox administration page. The application developer specifies information such as application context, display name, home URL and default saved state. Based on this information, DB2 Alphablox creates the application definition in the DB2 Alphablox repository and the application is made available to users through the application server.

The application context is a J2EE term referring to the descriptor that uniquely identifies the Web application or module. The application context serves as a container for J2EE applications that run within a J2EE server. Because they are standard J2EE applications, it is easy to package them as Web archive (WAR) or enterprise archive (EAR) files so they can be deployed to various application servers.

As specified in the J2EE standard, each DB2 Alphablox-enabled analytic application has a WEB-INF directory that houses the configuration information and the supporting resources necessary to deploy the application. These resources include components like Java classes (Java archive files) and JSP™ tag libraries.

The WEB-INF directory also includes the Web application descriptor file web.xml. The web.xml file, a standard file in all J2EE applications, is an XML file that contains markups that define the application behavior internally and as it relates to the application server. Included in the web.xml file are application-specific properties and their values, servlet mappings and security constraint information. This file enables the deployment into application servers, because it includes everything the application server needs to know. The application server reads the web.xml file during initialization of the application.

Managing metadata in the DB2 Alphablox repository

The Metadata Repository Manager controls the contents of the DB2 Alphablox repository. The repository is a database that holds application-specific metadata for applications and users. It also includes information on data sources, relational cubes, user groups, roles, applications and application states. When a user saves an application or Blox state, it is stored in the repository. The repository also stores bookmarked Blox properties that enable collaboration between users and groups, as well as XML representations of saved spreadsheet Blox.

DB2 Alphablox at run time

To support a widely dispersed user community, DB2 Alphablox provides a high degree of flexibility by allowing the developer or end user to choose the delivery format of DB2 Alphablox-enabled applications at run time. The same application can be deployed in different modes at different times to meet different requirements throughout the enterprise. This arrangement enables all users to leverage analytic solutions, regardless of any network bandwidth or client-side software limitations. It also allows applications to be optimized according to the application function and analytic capability required by the end user.

DB2 Alphablox application deployment options

Consider the following deployment scenarios, which are depicted in Figure 4-37:

- Static HTML: The application is delivered over an extranet or a narrowband network, providing users with simple data views. No significant client processing is required, and the information is presented in static HTML.
- Dynamic HTML: This mode utilizes JavaScript and cascading style sheet (CSS) to support the full range of data analysis functionality with a highly usable and customizable graphical user interface. It does not require any plug-ins or downloads of Java class files.
- XML rendering: The application data needs to be integrated with transactional application servers or delivered to clients such as cell phones or pagers. The application is rendered in XML for delivery to XML-enabled applications and clients.
- Ready for print: Users can request that pages be rendered for printing. The application presents the information, formatted for printing.
- Ready for PDF: The application user requires a report with greater control over page layouts, storage and printing. The application view is converted to PDF.

- Export to Excel or spreadsheet Blox: The application provides data to be analyzed in Excel or spreadsheet Blox, and exports the data to the chosen application.

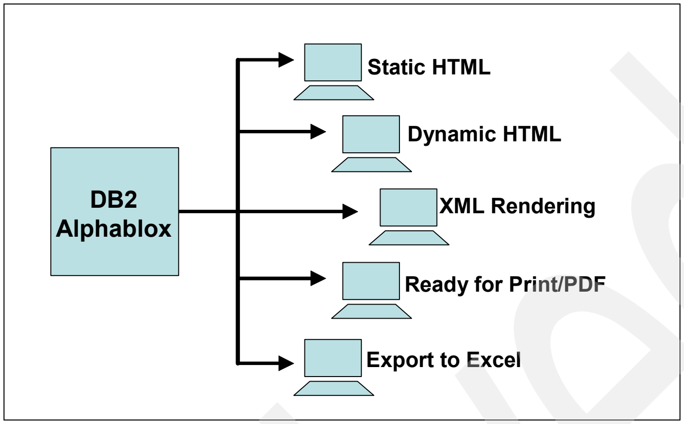


Figure 4-37 DB2 Alphablox scenarios

4.2.3 DB2 Alphablox services

In addition to application building Blox, the DB2 Alphablox platform consists of several services that help manage the applications, as seen in Figure 4-38. Each DB2 Alphablox service is responsible for a particular aspect of the application operating environment.

In this section, we take a closer look at each of these services.

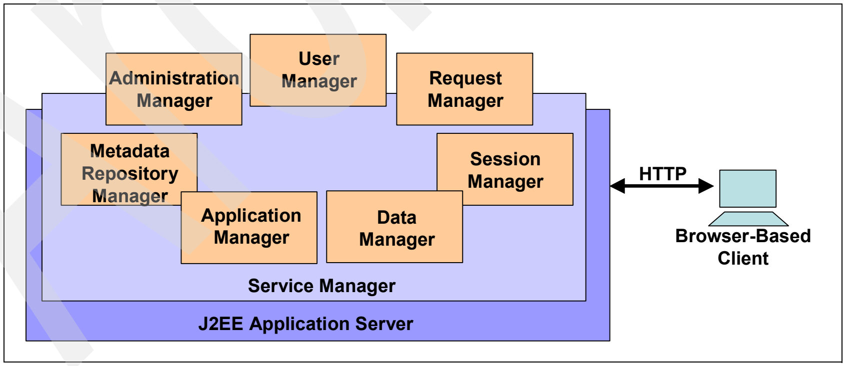


Figure 4-38 DB2 Alphablox services

Service Manager

As the focal point for server administration and monitoring, the Service Manager starts, stops and provides access to the other managers, passes service requests to the correct manager and monitors DB2 Alphablox resources.

Request Manager

The application server processes the original HTTP request; if there is DB2 Alphablox content, it is passed to the Request Manager for processing. If the request is from a user for which there is no active session, the Request Manager passes the request to the Session Manager. The Request Manager processes the application and Blox names, and then passes this information to the Application Manager for further processing.

As the application runs, the Request Manager coordinates communications between Blox on the application page and their server-side peers. The Request Manager also creates, monitors and manages threads for each request.

Session Manager

The Session Manager establishes a session for each new DB2 Alphablox browser instance. If an individual user has multiple DB2 Alphablox browsers open, the user would have multiple concurrent sessions. The Session Manager creates and manages session objects and tracks which applications a user visits. It also maintains a mapping between the DB2 Alphablox session and the session being maintained by the application server. The Session Manager also terminates dormant sessions after first saving the current state of each application and releases session resources.

User Manager

The application servers passes the user name to the User Manager, which gets the user information from the request object and then interacts with the application server through standard APIs to ensure that the user is authenticated. It controls all users of DB2 Alphablox services and creates and manages user instances. The User Manager also monitors resources allocated to each user and maintains user information, such as which applications are accessed, by which users, and for how long.

The DB2 Alphablox User Manager manages user authentication and authorization as well as provides personalization capabilities for customizing application content. By default, DB2 Alphablox uses its repository and the J2EE Security API to manage user and group information.

DB2 Alphablox also provides an out-of-the-box Lightweight Directory Access Protocol (LDAP) integration solution. This solution allows DB2 Alphablox to authenticate and authorize the users by using an LDAP directory server to recognize DB2 Alphablox users, groups and custom properties. The DB2 Alphablox User Manager is built on top of the personalization engine called the Extensible User Manager. For environments in which custom security is desired, the Extensible User Manager personalization engine provides interfaces that allow the extension of either of the out-of-the-box security solutions DB2 Alphablox repository or LDAP. It is also possible to plug in another external user manager such as NTLM or some existing Enterprise Java Beans (EJBs).

Application Manager

The Application Manager is responsible for creating or modifying the DB2 Alphablox application definition from the DB2 Alphablox administration applications page. The Application Manager verifies user privileges for application access, starts application instances, manages each application instance and supervises page processing before a page is presented to the user. The application design determines the exact page processing that occurs.

Application instance

The application instance controls the running state of each application. There is an application instance for each DB2 Alphablox browser instance in which the application is running. It is important to understand the difference between an application and an application instance, as depicted in Figure 4-39.

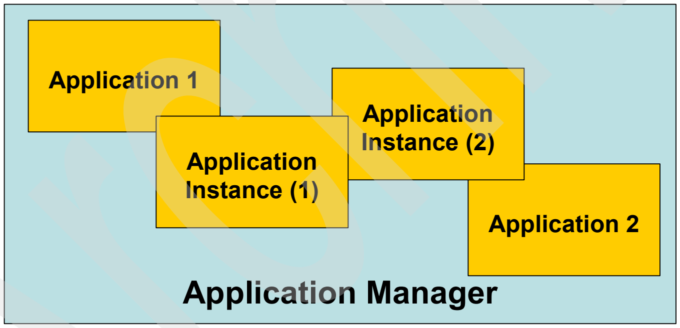


Figure 4-39 Application Services

An application is the entity JSP, HTML pages, images, servlets and so on that the application developer creates and stores on the DB2 Alphablox server. An application instance is the running state of that application, appearing in a browser window and interacting with a single user. The instance remains active until the client or administrator stops the application or the session times out.

The application instance maintains information about the state of each Blox in the application as well as information about the application as a whole. A user can choose to save the state of the entire application or simply that of an individual Blox. This feature can enhance collaborative analysis by enabling users to return to the saved state of the application and to share their results with others.

Data Manager

The Data Manager controls application requests for data sources, and is responsible for accessing, browsing, querying and retrieving data from data sources as depicted in Figure 4-40. It is uniquely designed for connectivity to a variety of data sources. Through a single API for both multidimensional and relational sources, the Data Manager translates the data into dimensions, rows, columns and pages the components typically used in multidimensional analysis. The Data Manager then presents this data for processing and manipulation by various Blox. Regardless of the data source, users perform analysis with the same analytical application front end.

The Data Manager architecture enables other databases to be connected through plug-in adapters. Each adapter encapsulates database-specific information for connection and processing. Data source definitions that identify the adapter are stored and administered centrally on DB2 Alphablox. If information for a data source changes, the application administrator changes the information in a single, central location.

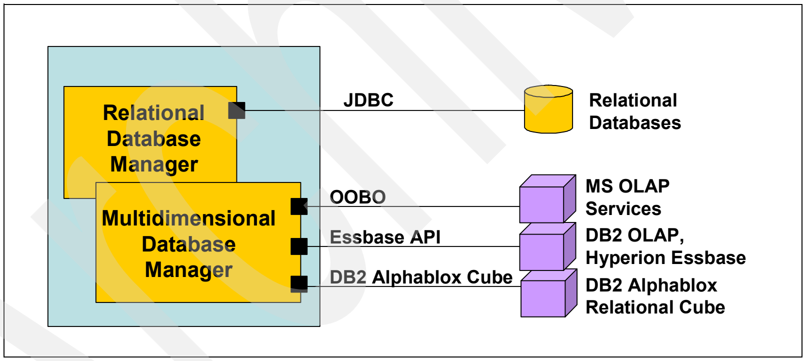


Figure 4-40 DB2 Alphablox data sources

The Data Manager and its associated data adapters provide support for:

- Browsing a collection of predefined data source connections, such as DB2 Alphablox named data sources
- Exposing the available databases within each data source
- Managing the database connections for user sessions
- Translating query objects into the underlying native query language
- Executing queries against the databases
- Modifying result set displays through user-invoked pivoting and drilling

- Write-back to the underlying data

In addition, the Data Manager allows for traversal of a result set and metadata. Users can retrieve data from the data source, traverse it using the DB2 Alphablox server-side result set and metadata APIs, and then take appropriate action. For instance, the application can be built to use the server-side APIs to traverse through the data looking for a certain condition based on an alert (for example, if actual inventory drops below plan). If the data meets the condition, a workflow process can be established that notifies the affected user (in this case the buyer for the product). The user can then write-back to the data source (order more product), closing the loop on the specific business process.

Content Manager

The Content Manager handles the setup of applications and examples that exist in the DB2 Alphablox Application Studio library of templates and tools. It has the ability to set up data sources and register applications.

4.2.4 Blox server/client structure

Each Blox has a server-side peer that contains the majority of the Blox functionality. Blox have the ability to render information to the client in a variety of formats. Server-side peers connect to a data source, obtain a result set and deliver it to the client in the requested run-time format. Client-side components and their server-side peers work together to provide data access, presentation and manipulation through the built-in user interface of the grid Blox and other presentation Blox.

Using server-side peers, as depicted in Figure 4-41, and client-side components, optimizes the performance of DB2 Alphablox analytic-enabled solutions. DB2 Alphablox manages the application logic, separating it from the user interface presentation, thus reducing the burden on the client machine.

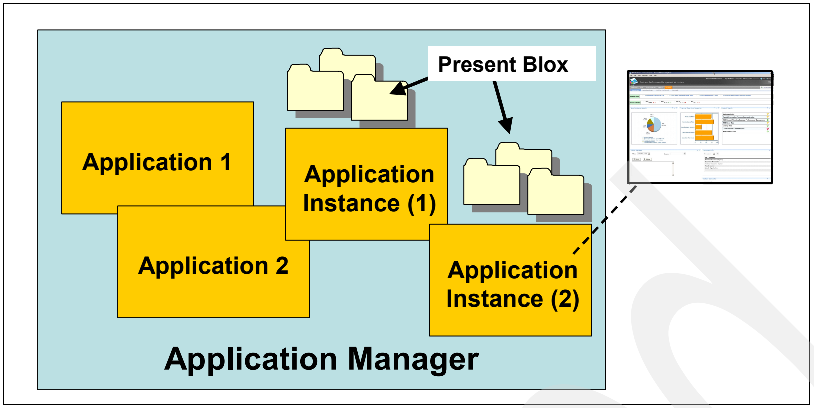


Figure 4-41 DB2 Alphablox server-side peers

Application delivery session flow

Numerous tasks are accomplished between the tiers as an application is accessed, and dynamically assembled and delivered to the client's Web browser. The page processes will vary based on the page type and content.

The application server is responsible for the following tasks:

- Network management
- Management of connections
- User authentication and security
- Processing and serving HTML files
- Processing and compiling JSP files using its servlet/JSP engine
- Serving the entire processed page back to the Web browser

DB2 Alphablox is responsible for the following tasks:

- Data access and manipulation
- Dynamically building and deploying the user interface that provides interactive analytic applications
- Managing the data session
- Personalizing the data view

Security

DB2 Alphablox leverages robust security models through the use of J2EE standard APIs (JAAS, JNDI). For example, if the application server uses LDAP, NTLM, Kerberos or another security mechanism, DB2 Alphablox will leverage that mechanism, as seen in Figure 4-42. In addition, DB2 Alphablox leverages the roles that are given to application server users.

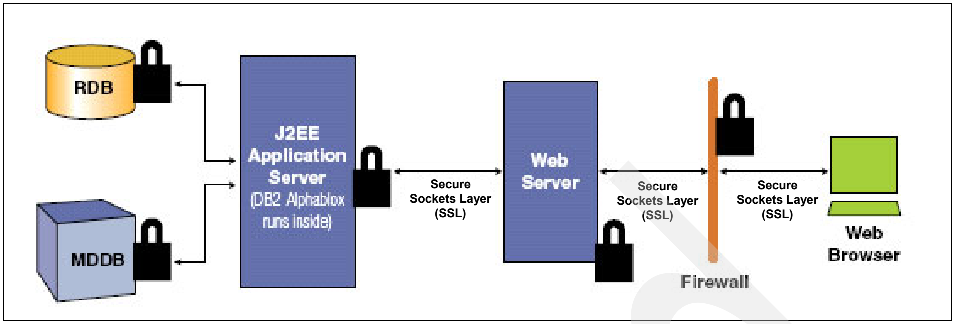


Figure 4-42 DB2 Alphablox security model

DB2 Alphablox users are managed via the application server administration, enabling them to be managed in the same way as users of other applications. This capability allows a developer to construct an application that uses the personalization capability from the application server, combined with information from DB2 Alphablox to dictate the content seen by a user.

By leveraging the application server security mechanisms, DB2 Alphablox developers can implement the security model that works best with their application server. In addition, DB2 Alphablox does not impose additional security constraints. For example, a DB2 Alphablox user ID can be generated by the username passed from the application server.

The only information passed to DB2 Alphablox from the application server is the username. Typically, usernames and passwords are also kept in the DB2 Alphablox repository to help enable personalization and to provide for database access. DB2 Alphablox supports and honors database security. Application users must provide valid database authorization to access application data. When the DB2 Alphablox password is synchronized with the database password, the user also can access the database without a separate sign-on to the database. DB2 Alphablox also works with Secure Sockets Layer (SSL) capabilities, if they exist on any of the servers. SSL encrypts and authenticates all messages sent between the browser, and the Web and application server pipe.

4.3 WebSphere Information Integration

An on demand business requires business processes, systems and people to be fully integrated within your company and with partners, suppliers and customers. Integration is at the heart of the on demand business.

As examples, we have described six of the critical business integration initiatives addressed by IBM WebSphere Information Integration, and have depicted them in Figure 4-43.

1. Master Data Management: Reliably synchronize all important business information dimensions, such as customers and products, across multiple systems.
2. Business Intelligence: Take the guesswork out of important decisions by consolidating trusted information in whatever form is needed, whenever it is needed.
3. Business Transformation: Transform companies into on demand businesses by isolating users and applications from the underlying information complexity.
4. Infrastructure Rationalization: Streamline corporate information access and reduce costs through an optimized information infrastructure.
5. Risk and Compliance: Deliver a dependable information management foundation to improve corporate visibility, ensure regulatory compliance and lower operational risk.
6. Corporate Portals: Provide information on demand while isolating users from the complexities of multiple data sources and application interfaces.

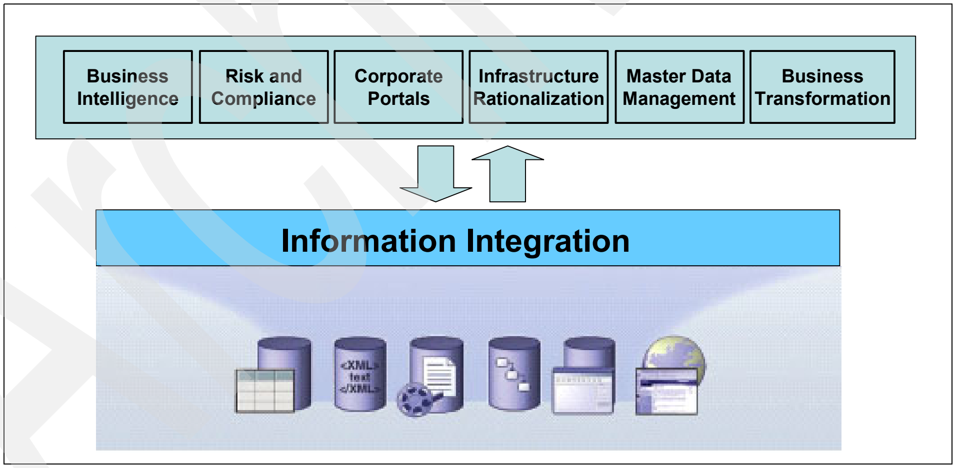


Figure 4-43 Business initiatives and Information Integration

As businesses replace manual processes with electronic processes and integrate processes across the enterprise, they must meet the challenge of finding, accessing, integrating, synchronizing and sharing a wide variety of information. WebSphere Information Integration software gives companies real-time, integrated access to business information structured and unstructured, mainframe and distributed, public and private across and beyond the enterprise.

Such information may reside in diverse source systems such as Oracle databases, SAP applications, Microsoft spreadsheets, flat files, the Web and news groups and be distributed across a variety of operating environments such as Microsoft Windows, Linux, UNIX and IBM z/OS® operating systems.

The software components that comprise WebSphere Information Integration are:

- WebSphere Information Integrator
- WebSphere DataStage
- WebSphere ProfileStage
- WebSphere QualityStage

4.3.1 WebSphere Information Integrator

WebSphere Information Integrator software offers a range of capabilities, such as enterprise federation, transformation, data placement (including replication and caching) and event publishing designed to meet varied integration requirements and easily integrate with industry-leading analytical tools, portal environments and packaged applications, application development environments, messaging-oriented middleware, service-oriented architectures (SOA) and business process software.

With these capabilities, business intelligence and business integration applications can find and access diverse and distributed information as though it were a single source, regardless of where it resides. Changes to information can be monitored in order to notify individuals or to trigger business processes. Moreover, administrators can more easily distribute, consolidate and synchronize information to facilitate application integration, maintain data warehouses and support business continuity across complex, multiplatform, multivendor IT environments.

IBM WebSphere Information Integrator integrates and transforms data and content. These capabilities are positioned on the IBM WebSphere Information Integration framework are depicted in Figure 4-44.

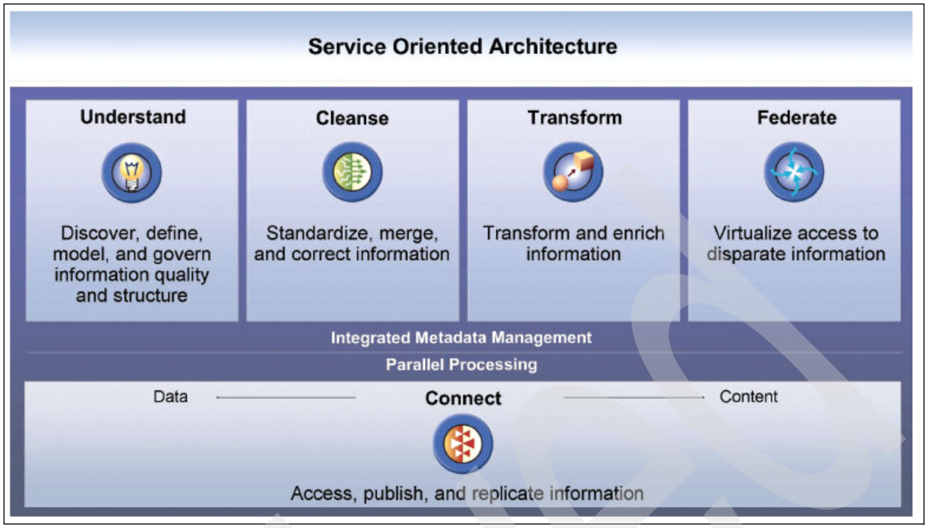


Figure 4-44 WebSphere Information Integration

The platform

The platform enables the following capabilities:

- Delivers accessible, consistent, timely and complete information.
- Provides leverage for businesses by allowing multiple types of business problems to be solved in an integrated manner, with a high degree of reuse and standardization.
- Supports validation, access and processing rules that can be reused across projects, leading to a high degree of consistency, control over data, and efficiency in IT projects, both in the initial development and over time as these rules need to be adjusted to meet changing business requirements.

The following sections outline the areas of functionality depicted in Figure 4-44.

Understand

Understand and analyze information, including its meanings, relationships and lineage. Businesses today deal with massive volumes of data often without much insight into the content, quality and structure of that data. Complex business transactions from customers and partners plus operational information moving within the enterprise are often the basis on which key business decisions are made. These decisions are often undermined by the lack of insight and understanding of the data. WebSphere Information Integration solutions provide the automated data profiling, analysis and monitoring capabilities to gather information about source data content, quality and structure.

WebSphere Information Integration solutions provide:

- Table and column data-driven analysis and reporting to help identify missing, inaccurate, redundant and inconsistent data.
- Data quality monitoring to help maintain the health of data throughout its life cycle.
- Automated discovery, and relationship and dependency analysis to establish the true metadata of the source systems.
- The ability to define, annotate, and report on fields of business data.

The product supporting these capabilities is IBM WebSphere ProfileStage, which is described in more detail in 4.3.3, “WebSphere ProfileStage” on page 181.

WebSphere Information Integration shares a common metadata foundation, allowing metadata to be shared and tracked across products. This approach to integration development results in faster implementation times and better collaboration between IT and business users.

Cleanse

The combination of customer, partner, and operational information provides the foundation for key business decisions made across the enterprise. The more error-laden these data streams and the more prolific the data, the less confident the decision makers are in using this information to drive the business. Business information needs to be clean: identified, standardized, matched, reconciled, and free of redundancies to ensure quality and consistency. Data cleansing enables the establishment of a single, logically correct view of core business entities across the enterprise, the foundation for master data management.

WebSphere Information Integration solutions provide:

- Standardization of source data fields, helping establish consistency in information.
- Validation, certification, and enrichment of common data elements, using trusted data like postal records for name and address information.
- Matching together of records across or within data sources, providing assurance that duplicate data is removed and enabling common entities from different sources to be identified and linked together.
- The ability to allow a single information record to be survived from the best information across sources for each unique entity, resulting in the creation of a single comprehensive and accurate view of information that spans across source systems.

These functions can be applied to any type of business entities, including customers, products, suppliers, employees, and chart of accounts. They are vital to improving information quality and enabling a comprehensive view of information about your most important business assets.

The WebSphere Information Integration product supporting data cleansing is IBM WebSphere QualityStage, which is described in more detail in 4.3.4, “WebSphere QualityStage” on page 185.

Transform

Whether matching an XML interchange specification, calculating key performance indicators or rendering complex content for viewing, a capability for transformation must accompany any integration offering.

WebSphere Information Integration software delivers rich transformation services. For basic data-type matching to complex analytical calculations, standard SQL provides a rich set of transformation features including functions such as string manipulation, arithmetic calculations, statistical computations, online analytical processing functions and procedural logic. Type-specific features—such as the application of scoring algorithms or chemical similarity searches—further enhance this already rich set of transformations.

Extensible Stylesheet Language (XSL) translations facilitate document interchange and dynamic style matching for diverse display characteristics. User-defined functions enable customers to standardize virtually any function for any data type. Stored procedures may encapsulate business logic for transformations and be programmatically invoked during replication processes. In addition, the ability to access Web services as a built-in function means that any Web service, such as a currency conversion, can become an embedded transformation function. Furthermore, content formats can be programmatically rendered into a browser-readable format.

WebSphere Information Integration solutions provide:

- High-volume, complex data transformation and movement functionality that can be used for standalone ETL or as a real-time data transformation engine for applications or processes.
- Embeddable in-line validation and transformation of complex data types, like EDI, SWIFT, HIPAA, and other semi-structured data formats.

The product supporting transformation is IBM WebSphere DataStage, which is described in more detail in 4.3.2, “WebSphere DataStage” on page 179.

Federate

WebSphere Information Integrator V8.2 software provides federation for enterprise information integration. The federation capability refers to the ability to allow applications to access and integrate diverse data and content structured and unstructured, mainframe and distributed, public and private as though it were a single resource, regardless of where the information resides, while retaining the autonomy and integrity of the data and content sources.

WebSphere Information Integrator software offers two complementary federation capabilities, one that offers an industry-standard SQL-based access paradigm to provide federation across a wide range of data and content sources and another that offers federation of content repositories, collaboration systems and workflow systems with an API optimized for the business needs of those who require broad content federation solutions.

WebSphere Information Integrator software has set the standard for federation with the following capabilities and characteristics:

- Transparency: helps mask the differences, idiosyncrasies and implementations of underlying information sources from the user, making the set of federated sources appear as a single system.
- Heterogeneous data access: enables federated access to highly diverse types of data and content.
- Extensibility: extends federated access to most data or content sources. Development and administration tools have been designed to minimize the effort required to integrate a new source, yet offer the flexibility to provide the information necessary to optimize query access.
- Rich functionality: includes standard SQL-supported functions, compensation for missing functions in back-end data sources, the ability to utilize source-unique capabilities and additional value-added capabilities beyond those natively delivered in the source systems. Content-centric federation includes the full range of common bidirectional operations plus value-added operations like single sign-on, security administration operations and federated search.
- Information source autonomy: enables federation of data sources with no impact to existing applications.
- Industry-leading performance: the SQL-based federation uses performance oriented capabilities such as cost-based distributed query optimization, parallel and anticipatory execution, native source access and transparent caching for great performance. Content-centric federation, although less performance-sensitive, utilizes native repository interfaces best performance.

Overall, the WebSphere Information Integrator federation capability helps developers write less integration code, allowing access to more information. It also provides a reusable information asset interface that can be leveraged as part of future applications and initiatives.

Federation via SQL-based access paradigm

WebSphere Information Integrator federation, via the SQL-based access paradigm, provides access to the entire range of enterprise data and content sources either directly or via interoperability between other WebSphere Information Integrator offerings. Using SQL and standard open database connectivity (ODBC) and Java database connectivity (JDBC) APIs, WebSphere Information Integrator software fits neatly and transparently behind common analytical and reporting tools, development environments, portals, ETL tools and other standard IT infrastructure components. SQL requests can be quickly and programmatically converted to Web services in an SOA. Alternatively, result sets can be programmatically converted into an XML document, validated and published with a single SQL request.

Data sources accessible via SQL-based access in WebSphere Information Integrator V8.2 software include:

- Relational databases: IBM DB2 Universal Database (UDB), IBM Informix Dynamic Server, Informix Extended Parallel Server, Microsoft SQL Server, Oracle, Sybase SQL Server, Sybase Adaptive Server Enterprises, Teradata and ODBC sources.
- Mainframe databases: VSAM, IMS™, Software AG Adabas and Computer Associates CA-Datacom and CA-IDMS, via separate purchase of WebSphere Information Integrator Classic Federation for z/OS.
- Packaged applications: SAP, PeopleSoft and Siebel via separate purchase of IBM WebSphere Business Integration Adapters.
- Content sources: Those accessible through the content-centric access paradigm described below in the content federation section via separate purchase of WebSphere Information Integrator Content Edition.
- Life sciences sources: Kyoto Encyclopedia of Genes and Genomes (KEGG) and data sources accessible by Entrez, BLAST, HMMER (including support for the HMMSEARCH tool) and BioRS.
- Other: WebSphere MQ message queues, Microsoft Excel spreadsheets, table-structured flat files, XML documents, data sources accessible via OLEDB and Web services, including complex XML results such as those providing access to earlier or older applications, content repositories or other data integration tools.
- C++ and Java developer toolkits to add access to other sources.

Content federation via content-centric access paradigm

Unstructured information is stored as large objects such as documents, correspondence, images and reports.

WebSphere Information Integrator federation, via the content-centric API, allows organizations to federate access to content stored across multiple disparate repositories, providing a single consistent interface to multiple systems. This allows organizations to easily build content-centric applications and enable existing applications for content with a single bidirectional interface for real-time views to content assets stored across the unstructured source domain, including content repositories, collaboration systems and workflow systems.

Just as with SQL-based federation, implementations using this method can access, manage and work with these repositories and systems as though the assets were stored and managed in a single unified repository. The content-centric API supports standard library services operations, such as check in, check out, versioning and annotations, and adds federation services, such as metadata mapping, federated search and single sign-on. Moreover, business process management is facilitated with the ability to monitor events in the content domain.

WebSphere Information Integrator software provides an out-of-the-box configurable Web client for knowledge workers and a single consistent Java and Web Services API for programmatic access from a single interface. Sources accessible via this content federation technology include DB2 Content Manager, DB2 Content Manager OnDemand, WebSphere MQ Workflow, Lotus Notes®, Lotus Domino Document Manager, EMC Documentum, FileNet Content Services, FileNet Image Services, FileNet Report Manager, FileNet Image Services Resource Adapter, FileNet P8 Content Manager, FileNet P8 Business Process Manager, Open Text Livelink, Microsoft Index Server/NTFS, Stellent Content Server, Interwoven TeamSite and Hummingbird Enterprise DM. In addition, a Java developer toolkit enables development, configuration and deployment of content connectors to additional commercial and proprietary repositories.

Overall, WebSphere Information Integrator SQL-based and content-centric federation capabilities are complementary. Used together, they provide transparent access and rich function to an extremely wide array of heterogeneous data and content sources.

The WebSphere Information Integration products supporting federation are:

- IBM WebSphere Information Integrator Standard and Advanced Editions
- IBM WebSphere Information Integrator Classic Federation for z/OS
- IBM WebSphere Information Integrator Content Edition

Connect

With businesses more distributed, consolidating, synchronizing, and distributing data across disparate databases is a core business requirement. The

WebSphere Information Integration portfolio meets these business demands, enabling businesses to connect to all their information. Solutions provide:

- Direct, native access to relevant sources bundled with each product for both mainframe and distributed computing environments.
- Consolidation, synchronization, and distribution across disparate sources.
- Support for a wide range of information sources, such as databases, files, content repositories and packaged applications.
- Changed data capture and event based publishing of data.

WebSphere Information Integration connectivity products can be used standalone to support specific application requirements or in conjunction with the other products in the platform to provide integrated composite solutions.

Replication

WebSphere Information Integrator V8.2 software offers replication that helps administrators distribute, consolidate and synchronize information across complex, multiplatform, multivendor IT environments. This software provides both queue-based and SQL-based replication architectures that present distinct benefits for various business needs.

Replication is used in a variety of contexts:

- Facilitate application integration. Whether point-to-point or distribution and consolidation topologies are required, it lets you manage data consistency between different application domains. For example, a retailer may replicate orders from showrooms to the production server and the latest inventory from the production server to showroom systems.
- Maintain data warehouses. Helps you utilize current information, capturing changes from transaction databases and replicating them into operational data stores, data warehouses or data marts to facilitate real-time business intelligence.
- Support business continuity. Can maintain synchronization for local or remote backup systems in either a standby or active mode.

Administrators can use a wizard-driven graphical user interface (GUI), command-line processor and script-driven processes to configure the variety of topologies, latency and consistency characteristics for both queue-based and SQL-based replication architectures. Integrated monitoring and reconciliation tools make it easy for administrators to react to problems and proactively maintain the health of the environment.

Queue-based replication

For DB2 Universal Database data sources and targets, queue-based replication architecture offers low latency and high-volume replication with managed conflict detection and resolution. Queue-based replication is designed to support business continuity, workload distribution and application integration scenarios.

Committed changes are captured from the database log and placed onto a WebSphere MQ message queue. A sophisticated apply process engine determines transaction dependencies and replays transactions on target systems to maximize parallelism and minimize latency. A set of conflict detection and resolution algorithms identifies conflicting updates from peer systems, allowing backup systems to work productively and application workloads to be distributed across multiple servers. In addition, data can be filtered so that only the data of interest is replicated, stored procedures can be invoked to facilitate transformations, and programmatic and high-performance load options can perform simultaneous loading and replicating.

SQL-based replication

For replication among databases from multiple vendors, WebSphere Information Integrator software uses an SQL-based replication architecture that maximizes flexibility in managing scheduling, transformation and distribution topologies for populating data warehouses or data marts, maintaining data consistency between disparate applications or efficiently managing distribution and consolidation scenarios among headquarters and branch or retail configurations.

In SQL replication, changes are captured with either a log-based or trigger-based mechanism and inserted into a relational staging table. An apply process asynchronously reads the changes from the staging table and handles the updates to the target systems.

Data can be distributed or consolidated via the replication server, and the data movement can be continuous, event-driven or automated on a specific schedule or at designated intervals. As with queue-based replication, data can be filtered so that only the data of interest is replicated. Moreover, standard SQL expressions and stored procedures can be invoked to facilitate transformations, and management of the data movement can be table-at-a-time for batch warehouse loading or transaction-consistent to maintain continuous online availability.

With SQL-based replication, you can replicate data between mixed relational data sources:

- Supported as sources and targets: DB2 Universal Database, Informix Dynamic Server, Microsoft SQL Server, Oracle, Sybase SQL Server and Sybase Adaptive Server Enterprise.

- Supported as targets: Informix Extended Parallel Server and Teradata.

Event publishing

WebSphere Information Integrator V8.2 software links data events with business processes, capturing database changes from mainframe data sources by reading the recovery log, formatting the changes into XML messages and then publishing them to WebSphere MQ. Any application or service that integrates with WebSphere MQ directly or supports Java Message Service (JMS) can asynchronously receive data changes as they occur. For example, using the event publishing capability, WebSphere Business Integration software can receive changes from a DB2 UDB database as they occur and can programmatically update an SAP application. Alternatively, a JMS-aware application or service within any Java 2, Enterprise Edition (J2EE) server (for example WebSphere Application Server) could receive those same changes and perform additional processing or integration. In addition, WebSphere Information Integrator event publishing solutions offer message formatting flexibility and allow changes to be filtered in multiple ways.

WebSphere Information Integrator event publishing solutions enable database events to initiate business processes. For example, a reduction in an inventory value could be used to drive a product restocking workflow, or the addition of a new customer could initiate a welcome e-mail, credit verification and an accounting update. This creates an application-independent, loosely coupled integration that is adaptable to changing application environments. For instance, while multiple applications may affect the value of the inventory level, a single point of integration the data items themselves is driving the workflow. Changes to the applications that affect the inventory level can be made with no impact on the event-driven integration.

Businesses can realize a faster time-to-market based on integration that captures the event in a single location, is more stable and is easier to maintain than integrations that attempt to monitor multiple applications for events.

Additionally, event publishing can deliver changes to ETL tools, custom-built processes for updating operational data stores or data warehouses that minimize bandwidth requirements and keep target databases more closely in sync. Thus, businesses can utilize current information for tactical and operational decision making.

Finally, event publishing solutions exist for content repositories accessible by the content-centric federation capability. Custom event handlers and other rules-based behaviors can span multiple content and workflow repositories. For example, a single event log can be established for centralized auditing in support of various compliance initiatives, or synchronization rules can be set up to migrate content as it is modified.

Event publishing is available from DB2 UDB for z/OS; DB2 UDB for Linux, UNIX and Windows; IMS; VSAM; and Computer Associates CA-IDMS sources.

The WebSphere Information Integration products supporting connect capabilities are:

- IBM WebSphere Information Integrator Replication Editions
- IBM WebSphere Information Integrator Event Publisher Editions
- IBM WebSphere Information Integrator Standard Edition
- IBM WebSphere Information Integrator Classic Federation for z/OS

4.3.2 WebSphere DataStage

IBM WebSphere DataStage, is a component of IBM WebSphere Information Integration platform. Whether you are building an enterprise data warehouse to support the information needs of the entire company, building a real-time data warehouse, or integrating source systems to support enterprise applications such as CRM, SCM, and ERP, WebSphere DataStage supports the enterprise data integration initiatives.

WebSphere DataStage supports the collection, integration and transformation of high volumes of data, with data structures ranging from simple to highly complex.

WebSphere DataStage can operate in real-time, capturing messages or extracting data at a moment's notice on the same platform that also integrates bulk data.

WebSphere DataStage delivers four core capabilities, all of which are necessary for data transformation within enterprise data integration projects:

1. Connectivity to a wide range of mainframe, earlier or older enterprise applications, databases, and external information sources to ensure that critical enterprise data asset can be used.
2. Intrinsic, pre-built library of 300 functions to reduce development time and learning curves, increasing data accuracy and reliability, and providing reliable documentation that lowers maintenance costs.
3. Maximum throughput from any hardware investment used in the completion of bulk tasks within the smallest batch windows, and the highest volumes of continuous, event-based transformations using a single high-performance parallel processing architecture.
4. Enterprise-class capabilities for development, deployment, and maintenance with no hand-coding required; and high-availability platform support to reduce on-going administration and implementation risk.

WebSphere DataStage is integrated with data profiling and data quality and cleansing products for scalable enterprise data integration solutions.

WebSphere DataStage can operate in real-time, capturing messages or extracting data on the same platform that also integrates bulk data. This allows you to respond to your data integration needs on demand. WebSphere DataStage transformation topology is depicted in Figure 4-45.

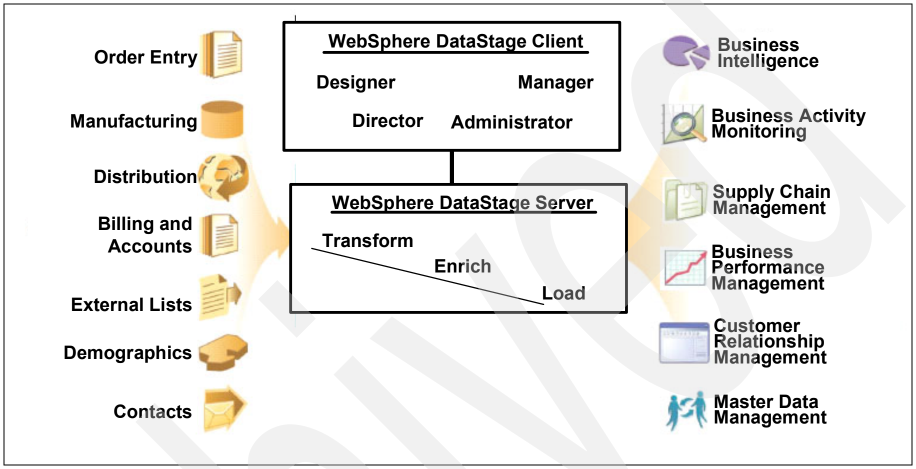


Figure 4-45 Transformation Topology

WebSphere DataStage supports a large number of heterogeneous data sources and targets in a single job, including:

- Text files.
- Complex XML data structures.
- Enterprise application systems including SAP, Siebel, Oracle and PeopleSoft.
- Almost any database, including partitioned databases such as Oracle, IBM DB2 Universal Database (with and without the Data Partitioning Feature), IBM Informix, Sybase, Teradata, and Microsoft SQL Server.
- Web services.
- SAS.
- Messaging and enterprise application integration products including WebSphere MQ and SeeBeyond.

Development environment

WebSphere DataStage employs a *work as you think* design metaphor. Developers use a top-down dataflow model of application programming and execution, which allows them to create a visual sequential data flow. A graphical palette helps developers diagram the flow of data through their environment via GUI-driven drag-and-drop design components. Developers also benefit from scripting language, debugging capabilities, and an open application programming interface (API) for leveraging external code. The WebSphere DataStage Designer tool is depicted in Figure 4-46.

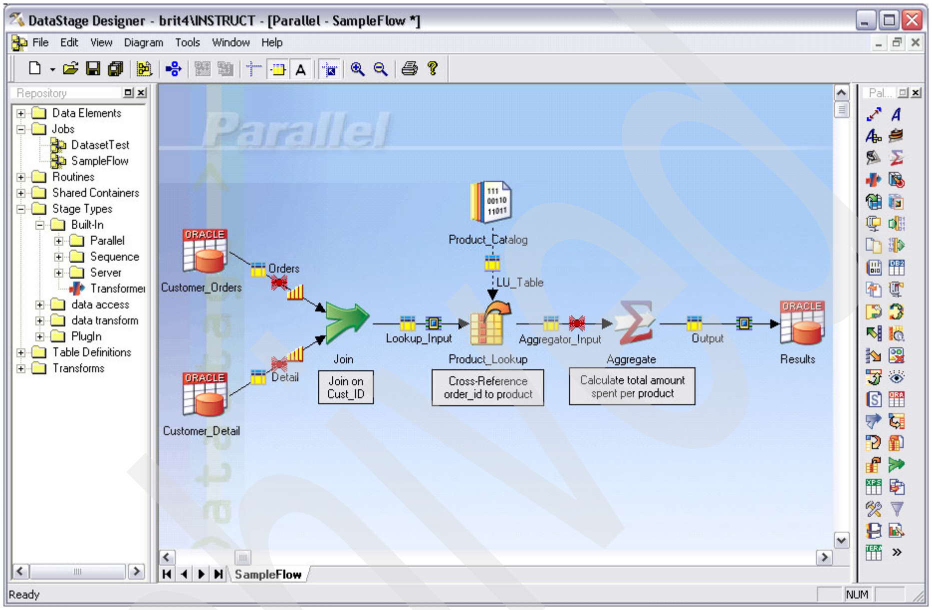


Figure 4-46 WebSphere DataStage Designer

4.3.3 WebSphere ProfileStage

WebSphere ProfileStage allows users to integrate multiple disparate systems by providing a complete understanding of the meta data, and by discovering dependencies within and across tables and databases. Because the meta data is based upon the actual source data, accuracy is nearly 100%, reducing the project risk by uncovering integration issues before development begins.

WebSphere ProfileStage brings automation to the critical and fundamental task of data source analysis - expediting comprehensive data analysis, reducing the time-to-market, and minimizing overall costs and resources for critical data integration projects. It profiles source data - analyzing column values and structures - and provides target database recommendations, such as primary keys, foreign keys, and table normalizations. Armed with this information, it builds a model of the data to facilitate the source-to-target mapping and automatically generates integration jobs.

Some of the functions/features of WebSphere ProfileStage are:

- Analyzes and profiles source and target systems to enable discovery and documentation of data anomalies.
- Validates the content, quality, and structure of your data from disparate systems - without programming.
- Enables metadata exchange within the integration platform.
- Provides a single and open repository for ease of maintenance and reporting.

No assumptions are made about the content of the data. The user supplies a description of the record layouts. Then WebSphere ProfileStage reads the source data, and automatically analyzes and profiles the data, so that the properties of the data (defined by the metadata) are generated without error. The properties include the tables, columns, probable keys and interrelationships among the data. Once these properties are known and verified, WebSphere ProfileStage automatically generates a normalized target database schema.

The business intelligence reports and source data to target database transformations are specified as part of the construction of this target database. After the source data is understood, it must be transformed into a relational database. This process is automated by ProfileStage, yielding a proposal for the target database that can be edited to get the best possible results.

The following is a description of the process and major components for profiling:

- Column Analysis: Here we examine all values for the same column to infer the column definition and other properties such as domain values, statistical measures, and min/max values. During Column Analysis, each available column of each table of source data is individually examined in depth. It is here that many properties of the data are observed and recorded, such as minimum, maximum, and average length, precision and scale for numeric values, basic data types encountered, including different date and time formats, minimum, maximum and average numeric values, count of empty values, NULL values, and non-NULL/empty values, and count of distinct values or cardinality.
- Table Analysis: This is the process of examining a random data sample selected from the data values for all columns of a table in order to compute the functional dependencies for this table. The purpose is to find associations between different columns in the same table. A functional dependency exists in a table if one set of columns is dependent on another set of columns. Each functional dependency has two components:
  - Determinant: A set of columns in the same table that compose the determinant. That is, the set of columns that determine the dependency. A determinant can consist of one or more columns.

– Dependent Column: One column from the same table that is dependent on the determinant. A column is said to be dependent if, for a given value of the determinant, the value of the dependent column is always the same.

- Primary Key Analysis: The process of identifying all candidate keys for one or more tables. The goal is to detect a column, or set of columns, which might be best suited as the primary key for each table. This analysis step must be completed before subsequent steps, such as Cross Table Analysis, can be performed. Normally, Primary Key Analysis uses results from Table Analysis. Table Analysis identifies the dependencies among the columns of a table, and records each as an aggregate dependency.

  Each row in the Aggregate Dependency table represents a single dependency for a given table. Each dependency has two components: a single column or set of columns (in the same table) that make up the determinant, and a set of columns (also in the same table) that are dependent upon the determinant. A set of columns is dependent on a determinant if, for a given value of the determinant, the value of the dependent columns is always the same. As you would then expect, a primary key determines all the values for the rest of the columns in the table. During Primary Key Analysis, one or more of the aggregate dependencies will become candidate keys. Subsequently, one candidate key must be confirmed by the user as the primary key.

- Cross-Table Analysis: This is the process of comparing all columns in each selected table against all columns in the other selected tables. The goal is to detect columns that share a common domain. If a pair of columns is found to share a common domain, then this might indicate the existence of a foreign key relationship between the two tables, or simply redundant data. These possibilities are examined during the subsequent Relationship Analysis step.

  Each row in the Domains Compared table represents a pair of columns whose domains have been compared during Cross Table Analysis. A domain comparison is a bidirectional process, which might conclude that one column's domain is contained within that of the other column, or vice versa. Each row in the Common Domains table represents the fact that one column (the base column) shares a common domain with another column (the paired column) in a different table. The common domain is noted only from the perspective of the base column; it makes no representation of whether or not a common domain also exists in the other direction.

- Normalization: Involves computing a third normal form relational model for the target database. The user interface provides a Normalization Wizard that guides the user through the process of normalizing the target database model. The information gained through the analysis phases is used to aid the user in making intelligent decisions in the construction of the target data model. When WebSphere ProfileStage spots a candidate for normalization, the user is presented with a proposed normalization. The user can accept the proposed normalization, reject the normalization, or modify the model.

- Reporting and Data Definition Language (DDL) Generation: The profiling reports describe in detail the information gained from the profiling steps. These reports can be used as the basis for estimating the scope of the project, for obtaining sign off from end users and stakeholders, and investigating the true composition of the source data. The reports can be output to the user’s screen, a printer, a file, email, Word and HTML, and a variety of other formats.

  The data model constructed can be exported to popular data modeling tools and in a variety of formats. The user can then examine the data model in a variety of combinations. If after examining the data model the user determines that changes in the target schema are necessary, values can be adjusted in the Normalization Wizard or in the analysis phases. New or revised models can be loaded into the WebSphere ProfileStage Metadata Repository and integrated into the project. WebSphere ProfileStage supports generation of SQL in a variety of dialects, including SQL Server, ANSI SQL and Oracle. The DDL can also be generated in XML format.

- Support for ETL Tools: Once the mappings have been confirmed, creating the ETL jobs for performing the creation of the target database is merely the push of a button. This approach also supports mapping from sources to predefined targets with a drag-and-drop interface.

  The code for WebSphere DataStage job transforms is automatically generated. An example of this is depicted in Figure 4-47. Here, a non-normalized source database in converted into a fully normalized target database. No programmer time was necessary to build the WebSphere DataStage jobs for these basic transformations. Since the WebSphere ProfileStage approach derives the data model for the target database from the information stored in the WebSphere ProfileStage Metadata Repository, the source-to-target mappings are automatically computed.

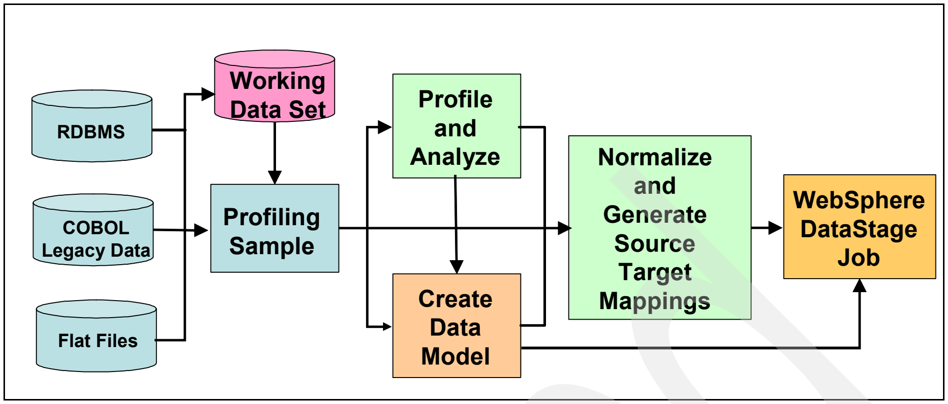


Figure 4-47 Data Profiling for data transformation

4.3.4 WebSphere QualityStage

WebSphere QualityStage helps your strategic systems deliver accurate, complete information to business users across the enterprise. Through an easy-to-use GUI and capabilities that can be customized to your organization's business rules, WebSphere QualityStage provides control over international names and addresses and related data such as phone numbers, birth dates, email addresses and other descriptive comment fields – and discovers relationships among them – in enterprise and Internet environments, in batch and real-time.

It analyzes data at the character level and uncovers anomalies and buried data prior to transforming it for database loading or transaction processing. Data from disparate sources is standardized into fixed fields using business driven rules to assign the correct semantic meaning to input data in order to facilitate matching.

Duplication and data relationships can be detected despite anomalous, inconsistent, and missing data values. There is a unique statistical matching engine that assesses the probability that two or more sets of data values refer to the same business entity - providing the most accurate match results available.

Once a match is confirmed, linking keys are constructed so users can complete a transaction or load a target system with true entity integrity and view related data as information. As a result, companies gain access to accurate, consistent, consolidated views of any individual or business entity and its relationships across the enterprise.

An example of a typical quality process is depicted in Figure 4-48.

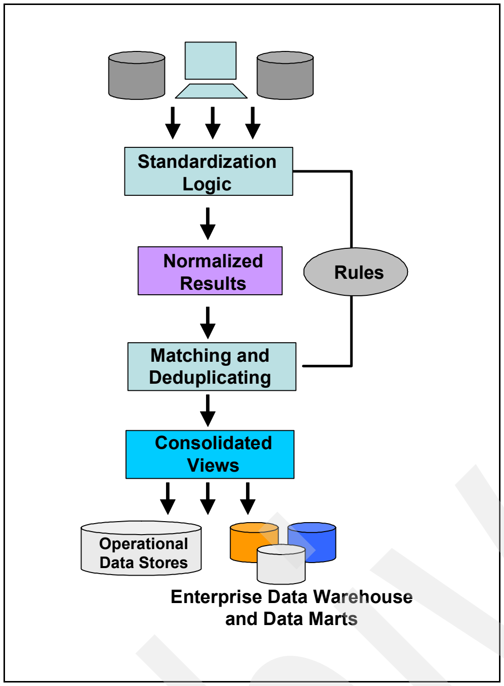


Figure 4-48 QualityStage Process

4.4 Process Integration

Process Integration enables disparate business resources — both inside and outside an enterprise — to work together to support the business strategy of the company. Model your enterprise business processes, perform activity-based costing, simulate the processes and deploy them. And then, while the processes are executed, monitor and manage them.

Process integration allows new business applications to be created and lets companies differentiate themselves by making customer transactions smoother. It can integrate the resources supporting business processes, automate the transactions between them and monitor those processes while executing them, making it visible, measurable and auditable.

Process integration returns control of business process back to business people. This means that line-of-business manager can change business processes an quickly adapt them to changing economic conditions.

Some of the products that comprise this solution are:

- WebSphere Business: Design, simulate and plan business processes
- WebSphere Business Integration Server: Provides an integration solution for process integration, workforce management, and enterprise application connectivity
- WebSphere Business Integration Server Foundation: Provides a next-generation application server that simplifies build-to-integrate tasks, accelerates application development, and enables dynamic application flexibility
- WebSphere Business Monitor: Optimize performance against business measures
- WebSphere Business Integration Connect: Enables operational B2B based on communities of trading partners

Business performance management is an initiative that enables companies to model, develop, and monitor business processes as a proactive way to really begin to monitor and manage their business.

Lets take a closer look at a few of these key products.

4.4.1 WebSphere Business Modeler

WebSphere Business Modeler products help organizations fully visualize, comprehend, and document their business processes. Rapid results can be obtained through the collaboration functionality, where subject matter experts team to define business models and eliminate inefficiencies. You can model business processes, deploy, monitor, and take actions based upon key performance indicators (KPIs), alerts, and triggers for continuous optimization. Business processes then get tightly linked with strategic corporate objectives and honed as required. WebSphere Business Modeler products can drive much more granular business insight and knowledge, where knowledge equates to competitive advantage.

WebSphere Business Modeler products can serve to close the gap that exists between organization business units, and IT understanding of the business drivers. IT requirements for defining a system will be well articulated, since both are utilizing a common framework, implemented through an Eclipse 3.0 shared workspace. Given that a business process is a defined set of activities leading to specific results, modeling them assures that your best practices are well documented and communicated. With this modeling and simulation tool, those who best understand the business can model critical business processes. Accurate business modeling is the starting point for successful IT deployment.

WebSphere Business Modeler helps you model those aspects of the business that matter the most. The business measures editing function allows you to define the KPIs and metrics for your organization. When modeling your information assets, the information model will provide a view of your data, and its exact use within a business process. The resource model allows you to identify all of the different resource types so they can be associated directly to the model. The swim-lane view can be used to visually display the model according to role, resource, organization unit, and classifier. You can both view and edit models in the swim-lane view.

WebSphere Business Modeler products are core components of business innovation and optimization (BIO). BIO enables you to understand the status of business processes across business and IT, put that understanding in context against goals and trends, and then take action quickly to improve execution. The Modeler solution delivers a structure for continuous business improvement. A core underlying functionality is the ability to analyze and model business processes. WebSphere Business Modeler provides software tools to help you model, simulate, and analyze complex business processes. Line-of-business managers as well as business analysts can design and develop process flows that improve how you do business. With the WebSphere Business Modeler products, you will be able to maximize business effectiveness by optimizing processes that give you competitive leadership in today's on demand business environment.

The four products in the V6 WebSphere Business Modeler family are:

- WebSphere Business Modeler Basic: provides a low-cost option for business users who are looking for a simple and easy-to-use tool to model, document, and print their business processes.
- WebSphere Business Modeler Advanced: provides all the features of WebSphere Business Modeler Basic, plus complex model simulation and analysis capabilities. In addition, it lets IT-focused users export models to multiple build time environments to help jump-start application development.
- WebSphere Business Modeler Publishing Server: provides the ability to publish business process models to a Portlet-based server where multiple subject matter experts can view and review the information simultaneously through a standard Internet browser.
- WebSphere Business Modeler Publishing Edition: consists of 10 WebSphere Business Modeler Advanced licenses, plus one license of the WebSphere Business Modeler Publishing Server.

WebSphere Business Modeler Advanced

WebSphere Business Modeler Advanced bridges the gap between line of business and IT. It provides functionality for process modeling, enterprise modeling, essential data and artifact modeling, organization modeling, resource modeling, timeline and location modeling, and business process analysis. The Modeler can be used in five modes:

- Basic
- Intermediate
- Advanced
- Business Process Execution Language (BPEL) for either the WebSphere Business Integration Server Foundation or WebSphere Process Server runtime
- WebSphere MQ Workflow Flow Definition Language (FDL)

With WebSphere Business Modeler Advanced, business and IT communities utilize an Eclipse-based shared workspace. BPEL and FDL allow you to work in the mode with which you will be deploying. Business analysis is core for improving businesses. Many business process modeling efforts stop at developing flow diagrams or process maps. With WebSphere Business Modeler Advanced, the effort is extended to include simulation, analysis, and redesign.

WebSphere Business Modeler Advanced has a visual process modeling environment that is enhanced by the ability to color code the elements by role, classification, or organization units. In addition to the name of the object, labels can be added to the top and bottom of an object. User customization allows the appropriate labels to be displayed on the model at the appropriate time. In addition to these visual analysis techniques, a swim-lane view can be used to display the model according to role, resource, organization unit, and classifier. Modeler V6 provides the ability to view the model via swim-lane and also edit the model in the swimlane view.

WebSphere Business Modeler Advanced includes a simulation engine that allows you to simulate the behavior of your process, permitting analysis of workloads and bottlenecks. You can view analysis on the process, resources, activity, and queue during the simulation or after the simulation is complete. The flows are animated in a step-by-step simulation that allow you to see real-time data on the flows. Simulation snapshots are available for reference. The simulation engine enables you to determine the most efficient model prior to implementation by quickly conducting the "what-if" analysis. To quickly go from the modeling phase to the simulation phase, metric information (such as costs and time) are entered into the model while it is being modeled in the process editor.

WebSphere Business Modeler Advanced provides capabilities for adding and viewing simulation attributes in a table format to simplify the steps to adding the attributes and running a simulation. In addition new distributions have been added to support additional statistical types of analysis that have been widely used in the healthcare industry but are applicable across industries. Another important feature to support the simulation capabilities is the ability to generate multiple instances of a single resource to faster simulate resource utilization within a business process.

WebSphere Business Modeler Advanced includes a reporting tool that can automatically create written, numerical, and graphical reports. The reports provide guidance in the process analysis and redesign. Some of the predefined reports are:

- Process Summary Report: provides a single report that contains essential cost and time analysis reports.
- Process Comparison Reports: combines and compares the Process Summary Reports from two process simulations for comparisons and provides ROI comparisons of as-is and to-be flows.
- Documentation Report: provides an overview of the attributes of business items, resources, or other model elements.
- Procedure Report: documents the sequence of steps within a process, and the relationships of that process to other processes and roles.

All of this can be rolled up into communication tools through charts, tables, reports, graphs, or diagrams. If a customized report is needed, there is a Report Builder to help you generate it which also supports publishing models to the Web.

Modeler V6 includes integration with Crystal reports enabling you to create reports in Crystal reports that can combine business process information with additional business information.

Once the models have been built, simulated, analyzed, and corrected, you can deploy them to multiple code generation tools. The models can be exported to WebSphere Studio Application Developer Integration Edition V5.1 through Business Process Execution Language (BPEL), Web Services Description Language (WSDL), and XML schema definitions (XSDs). In addition WebSphere Business Modeler exports to the new WebSphere Integration Developer tooling environment through the generation of Service Component Architecture (SCA) components, modules, task execution language (TEL) for human tasks, BPEL, WSDL, and XSDs. In addition WebSphere Business Modeler Advanced can create the wiring of those artifacts. WebSphere Business Modeler Advanced also supports the capability to export to Rational XDE™, under the Business Modeling Profile. Integration with Rational Software Architect (RSA) is performed within the RSA environment by importing the WebSphere Business Modeler process models into the RSA tool. And finally, the tool still supports integration with WebSphere MQ Workflow through flow definition language (FDL). FDL can be exported directly to WebSphere MQ Workflow Build-time.

In addition to the deployment capabilities, WebSphere Business Modeler provides model management capabilities by being able to version, check in and check out, merge, and compare different versions, track element history, and get a specific version from the history. These capabilities are provided by the CVS plug-in as part of the Eclipse platform. WebSphere Business Modeler offers optional support for Rational ClearCase® as a version control system.

Another important feature of WebSphere Business Modeler is the ability to generate business measures models via the business measures editor. This feature enables Modeler to support WebSphere Business Monitor by allowing you to take a snapshot of a modeled process and update it with key performance indicator (KPI) and metric information. KPIs and metrics are modeled by defining what triggers them and under what conditions they are monitored. In addition situations can be defined that allow WebSphere Business Monitor to determine a situation has occurred and an action must be taken. Once you have defined the KPIs, metrics, and situation, an integration specialist can go in and ensure that the appropriate events both are available and contain the correct information in order to properly be monitored. Once this has been completed, a business measures model is generated. This business measures model is a deployable artifact for the WebSphere Business Monitor.

WebSphere Business Modeler Advanced is able to import FDL from WebSphere MQ Workflow, models from WebSphere Business Integration Modeler V4.2.4, and WebSphere Business Integration Modeler V5 models. Included with the product are Visio® import capabilities as well as a proprietary XML import and export capabilities. This XML capability can be used to facilitate integration with other tools, allowing users to import information from external sources, as well as export it to additional environments.

Modeler allows you to support and implement enterprise initiatives, such as:

- Workflow automation
- Policy and procedure documentation
- Application development
- Sarbanes-Oxley initiatives
- HIPAA

WebSphere Business Modeler Advanced provides support for:

- WebSphere MQ Workflow by exporting in FDL
- WebSphere Server Foundation by exporting in BPEL
- WebSphere Process Server by exporting BPEL
- Rational XDE through UML 1.4 export

WebSphere Business Modeler Basic

WebSphere Business Modeler Basic is a low-cost entry edition providing basic modeling to capture data. The Basic Edition is for enterprise departments and individual users who do not require the full capabilities of WebSphere Business Modeler Advanced. WebSphere Business Modeler Basic is used for modeling, validation, and documentation. All three modeling modes (basic, intermediate, and advanced) are available. Once the models are entered, the validation rules still apply; however, WebSphere Business Modeler Advanced is needed to utilize the simulation engine and analysis. The reporting tool and the complete set of static documentation reports are available in this base release. The models can be captured, validated, and documented in this version. WebSphere Business Modeler Advanced is needed to run simulations and export to the multiple code generation tools.

Table 4-4 describes the key functional differences between WebSphere Business Modeler Basic and WebSphere Business Modeler Advanced:

Table 4-4 Basic and Advanced comparison

feature	Basic	Advanced
Modes	Basic, Intermediate, Advanced	Basic, Intermediate, Advanced, BPEL, FDL
Versioning	Yes	Yes
Simulation	No	Yes
Static and Dynamic Analysis	No	Yes
Reporting (create reports)	Yes	Yes
Query Report	Yes	Yes
Basis Report Templates	Yes	Yes
Printing	Yes	Yes
Modeler Project Import/Export	Yes	Yes
Delimited file Import/Export	Yes	Yes
ADF Import	Yes	Yes
XSD Import/Export	No	Yes
UML Export	No	Yes

feature	Basic	Advanced
FDL and BPEL Export	No	Yes
FDL Import	No	Yes
Visio Import	Yes	Yes
XML Import/Export	No	Yes
Swimlane Editor	Yes	Yes
Business Measures Modeling	No	Yes

WebSphere Business Modeler Publishing Server

WebSphere Business Modeler provides the capability to publish business processes created within WebSphere Business Modeler. Once published, authorized viewers can view and comment on the business processes using a standard Internet browser.

Publish the business process: WebSphere Business Modeler is able to create visual representations of business processes along with supporting information. The more accurate and detailed the information contained in WebSphere Business Modeler, the more accurate the analysis. An important step in modeling the business processes is to validate the process and its data. One way to achieve this validation is to make the business processes available so that subject matter experts and other interested parties can review them. Publishing Server allows someone using WebSphere Business Modeler to publish an entire business process modeling project or just parts of it to a server. Models are published in the standard process model view, not the swimlane view. The subject matter experts and other reviewers can then view the process diagrams and their supporting information using a standard Internet browser. By more widely publishing business processes, you increase the likelihood of the model being accurate and reduce the risk of a subject matter expert introducing information that is unique to that user.

A published business process can also serve as training tool for new employees. The business process model can provide new employees with a high-level view of their role in the company and clearly identify the tasks they are expected to perform. It can identify what the employees need to do in certain conditions, if the model has that level of detail.

In addition to just viewing business processes, authorized reviewers can use Publishing Server to comment on a diagram or on its supporting information. Because the comments are visible and other reviewers can respond to the comments, Publishing Server provides a forum for discussing and resolving differences of opinion on a business process. The comments and responses are tracked by originator and date and time at creation. These comments and responses can then be exported permitting the modeling team to update the model, making it more accurate.

WebSphere Business Modeler Publishing Server also allows authorized reviewers to post attachments such as MS Word documents. The attachments allow the reviewer to provide the modeling team with additional information that they can use to update the business processes.

WebSphere Business Modeler Publishing Server has three major components: publisher, publishing server, and client.

Publisher

The publisher is a plug-in for Business Integration Modeler that takes modeling projects or selected elements from a modeling project and publishes them to the publishing server.

The publisher acts as a gateway to the publishing server. The person performing the role of publisher selects which projects or model elements to publish and which publishing server will host the published project. Projects or model elements can be published in an in-progress mode. An administrator can then publish the information into a released mode. Information published in both modes is displayed with information contained in the advanced mode of WebSphere Business Modeler. This becomes useful when working with multiple versions of the business process models so that the correct level of information is shown to users at the right time.

Publishing server

The publishing server hosts published business modeling projects. The server consists of WebSphere Portal Server and a database. WebSphere Portal Server displays the business process information in four portal windows and an additional portal for displaying comments. The database stores the comments.

The publishing server also has an administration page that allows an authorized person to perform the administration role to define user access, edit and remove business processes, and edit and remove comments.

Client

The client is a standard Internet Explorer® Internet browser with the Scalable Vector Graphics (SVG) Viewer plug-in. The SVG Viewer enables you to zoom in and out of the process diagrams.

To access a published business process model, the person performing the role of reviewer enters the model's URL on the publishing server. Once the reviewer has logged onto the publishing server, the client displays the portals served by the server. The information displayed by the portals depends on what elements or buttons the reviewer has clicked in any of the portals.

When viewing published business process models, you can select information either through the navigation portlet or the view portlet, once a process or element has been selected. Once a business process model or model element is selected, you can view its associated attribute information as well as any comments or responses or add new comments and responses via the comments portlet. In addition you can associate additional read only documents or URLs to model elements in the view portlets to help provide additional contextual information about the element.

4.4.2 WebSphere Business Monitor

The WebSphere Business Monitor Version 6.0 is a Web-based client/server application that measures business performance, monitors processes and workflow, and reports on business operations. It monitors business processes at runtime by monitoring event-emitting runtime engines. The information captured can help you identify problems, correct faults, and change processes to achieve a more efficient business.

WebSphere Business Monitor calculates key performance indicators (KPIs) and metrics using collected events, based on a given model. The calculated KPIs and metrics values are represented on a number of views based on business needs. Users are notified of incidents requiring their attention and corrective actions can be performed to avoid failures. It supports different notification methods (such as alert, e-mail, cell phone, pager, and service invocation) against situations and actions associated with defined conditions.

WebSphere Business Monitor includes one copy of WebSphere Business Modeler Advanced providing modeling and simulation capabilities to model the critical aspects of businesses. WebSphere Business Monitor is used in conjunction with WebSphere Business Modeler to create business measures model that specifically identifies activities to be monitored, including events, business metrics, and KPIs.

Business measures models are used for monitoring procedures. These models are created in the Business Measures editor where you can specify the measuring points and event filters, define the measurements, their correlations, and sources of the business data. When the business measures model is complete, you can export it to WebSphere Business Monitor. It then recognizes the model to be monitored and the measurements to be captured from the incoming events.

The business measures editor is used to open the process models created in WebSphere Business Modeler and to create business measures models. For each business measures model, you can define the metrics and KPIs, event emission points, event filters, event composition rules, and situations that will trigger specific actions at runtime.

Once the business measures model is complete, work can be performed to enable it to be recognized by WebSphere Business Monitor. Then the Business Monitor understands the measurements to be captured from incoming events.

In addition, continuous business process improvement metrics, such as task working durations and process decision percentages are calculated and can be exported to update their corresponding business process models in the WebSphere Business Modeler. This improves simulation and analysis results because now the actual metrics (versus assumptions) are factored into the model. These capabilities provide for processes running in WebSphere Process Server, which is described in more detail in 4.4.7, "WebSphere Process Server and Integration Developer" on page 208.

Many business process modeling efforts stop at developing flow diagrams or process maps. With WebSphere Business Modeler Advanced, this effort is extended to include simulation, analysis, and redesign.

4.4.3 Architecture

The architecture of WebSphere Business Monitor 6.0 comprises a set of internal components and a group of external components.

The diagram shown in Figure 4-49 represents the overall logical reference architecture and the components of Monitor V6.

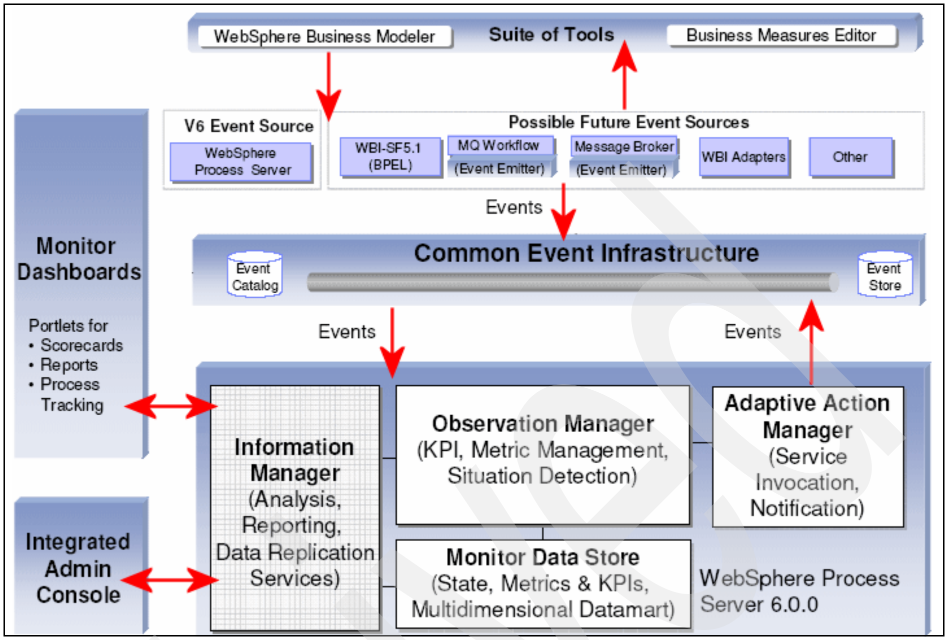


Figure 4-49 WebSphere Business Monitor architecture

Internal components overview

The following is a list of the internal components:

- **Monitor server**—Receives events, handles monitoring-context instances, and stores and persists runtime and historical metrics and KPI values of those instances.
- **Dashboards**—Display the monitored data. They provide a predefined set of views that can be customized to support different representations of data and offer enhanced data analysis.
- **Schema generator**—Generates database scripts to be used for creating databases tables in state, runtime, and historical databases. These databases contain the business measures models data. The schema generator also generates the DB2 Cube Views metadata description of the historical database and generates the metadata mappings for the replication manager.
- **Databases**—Provide the Monitor server with information for event processing. They also provide the dashboard client with information for populating views. Information is transferred across the databases through another monitor component, the replication manager.

- **Adaptive** action manager—Provides different types of business responses resulting from situations expressed within the incoming events.

External components overview

The following is a list of the external components:

- **Business measures editor** (BME)—It is used to create the business measures model that defines what should be monitored, for example, monitoring contexts, key performance indicators, metrics, and business situations.
- **Common event infrastructure** (CEI)— WebSphere Business Monitor uses the Common Event Infrastructure (CEI) and the Common Business Event (CBE) format. This means that the WebSphere Business Monitor leverages the CBE format for consuming and emitting events.

  It participates in event management by receiving events from event sources and transferring them to the event consumers that have expressed interest in those events.
- **DB2 Alphablox and DB2** Cube Views—Provide enhanced data analysis for dashboards.

Figure 4-50 summarizes the overview of the components.

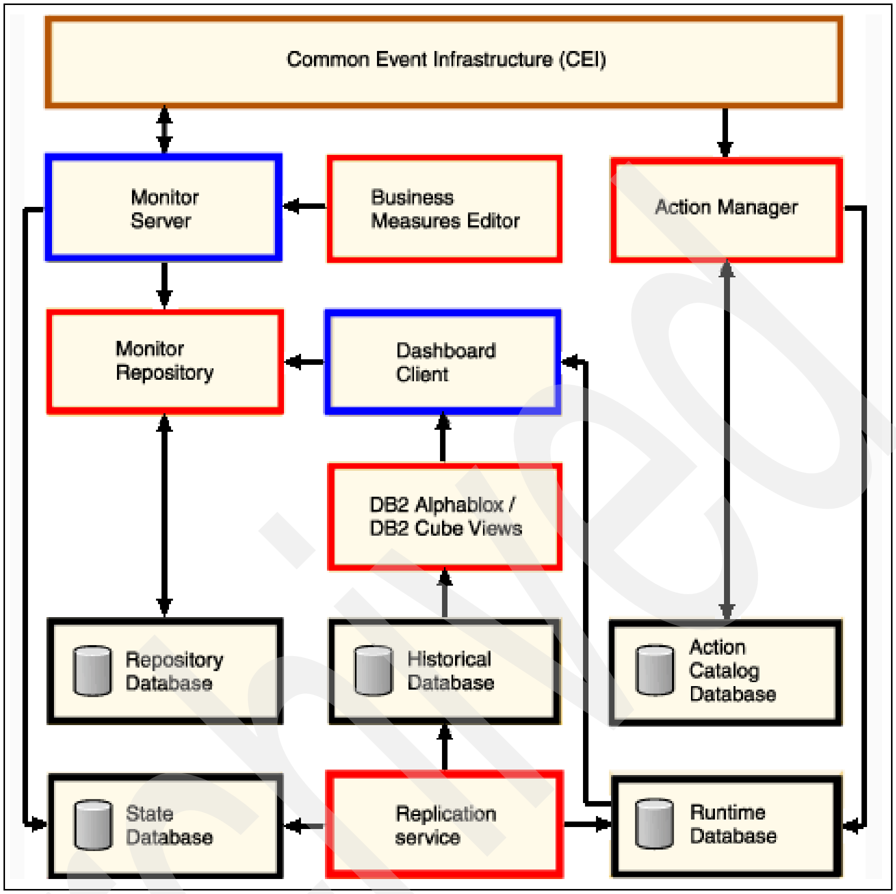


Figure 4-50 WebSphere Business Monitor component overview

4.4.4 Component details

WebSphere Business Monitor 6.0 enables you to monitor the run time behavior of business processes through a Web application that will be deployed on WebSphere Process Server v6.0. The data that will be monitored is issued from a runtime engine.

Business measures model

Monitoring of data is based on a business measures model which includes artifacts that permit correlation of the runtime events with a specific instance, in addition to entries that specify situations. The business measures model is obtained from the original business model by editing entries that are essential for monitoring purposes: correlation of events, metric calculations, detecting situations. Editing of the model is done by the Business Measures Editor.

Event handling

Data is encapsulated in common base events (CBE) by means of event emitters and are transmitted on a common event bus, the common event infrastructure (CEI). For Monitor V6, only events emitted from WebSphere Process Server are supported.

Monitor V6 runs on WebSphere Process Server V6, which sits on top of WebSphere Application Server V6. The runtime engine (WebSphere Process Server) for the real application runs on another machine and CBEs are extracted from the engine and submitted on the event bus, configured as a service integration bus (SIB) between the two machines. The Monitor Server receives the CBE events from the SIB and matches them to monitoring context instances and calculates the appropriate metrics and KPIs to ultimately be displayed for the user on the dashboard.

The common event infrastructure (CEI) is a shared component that can operate either inside or outside WebSphere Application Server. The CEI provides facilities for the run-time environment to persistently store and retrieve events from many different programming environments. Events are represented using the common base event model, a standard, XML-based format that defines the structure of an event. The events are passed through JMS across the service integration bus (SIB).

WebSphere Business Monitor server

WebSphere Business Monitor server is used to perform the management of the KPIs and metrics, as well as other relevant information from the business measures model which is defined as part of the business measures modeling capabilities of the WebSphere Business Modeler. The monitor server is also responsible for event correlation as well as situation detection that then triggers the adaptive action manager component.

Adaptive action manager

The adaptive action manager (or action manager) is another key component in the Monitor 6 product architecture, which provides the *sense and respond* functionality. The adaptive action manager is the WebSphere Business Monitor component that receives situation events emitted by the observation manager. It selects appropriate actions based on predefined bindings between the situations and actions that are set by the user, and invokes one or more action services. The CEI sends these situation events to the adaptive action manager, which parses them, selects appropriate actions based on predefined rules created by the user, and invokes a selected action or set of actions.

The adaptive action manager performs two types of actions: notification actions and service invocation actions:

- Notification actions take the form of e-mail, SMS, pager message, or a dashboard alert.
- Service invocation actions invoke a Web service, or a BPEL process through a Web service invocation.

The adaptive action manager parses the received situation event and selects an appropriate action by looking up the action in the action catalog database, where action-related information and binding information are stored. If the appropriate action is a dashboard alert, the action manager extracts the data needed for the creation of the alert-notification record from the received situation event and inserts this record in the WebSphere Business Monitor Runtime database. The record is collected by the Alerts view in a dashboard. The adaptive action manager uses LDAP as the user-registry for user notifications.

4.4.5 Databases

The data architecture of Monitor V6 has been optimized for both transaction processing data stores as well as data marts for reporting, analytics and business intelligence.

Simply stated, the Monitor V6 is responsible for its own data store to handle data required for the monitoring operation: instances of running monitoring contexts and metric values. The performance is optimized by dividing the data store into different databases, each being optimized for specific types of DB access operations.

DB2 replication services are responsible for moving state data to the historical data store at configurable replication intervals. This fundamentally separates the transaction processing data store from the historical data store for high-performance event processing.

Data analysis can then be performed on the historical data, made available by introducing DB2 Cube Views and accessing cubes from DB2 Alphablox interface, which is the visualization module.

The database topology for the Monitor Server and Dashboard Server in a given server environment may vary, for example:

- The Monitor Server runs on its own machine with the State and Repository databases.
- The Monitor Dashboard runs on a separate machine using WebSphere Application Server and Portal Server with the Runtime and History databases.

This setup is done for performance reasons. However, you may want the Repository database on the Dashboard Server because the Monitor Server only uses the Repository database at the time that you import a model. However the Dashboard Server uses it frequently when configuring and displaying the dashboards.

The Monitor uses a number of databases to store event information. Here is a short description of the databases.

Repository database

The Repository database contains the metadata describing the currently deployed business measures models as well as information about the other Monitor databases. The Repository database contains the history of the deployed models. There is only one Repository database per Monitor installation.

The Repository database is used by the Launchpad, which populates it with the database attributes for the State, Runtime, and Historical databases. These attributes are the database name, database schema, and host names of the database server. They are used by the other Monitor components to access the State, Runtime, and Historical databases at runtime. The Repository database is also populated during the import of the business measures model.

State database

The State database stores information about running instances. This information includes metrics, business measures, and key performance indicators (KPIs) values. It is optimized for heavy transaction workloads. There is only one State database per Monitor installation.

Each process instance requires two tables in the State database to store metrics, business measures, and KPIs. The structure of these tables is as dynamic as the structure of the process instance. Each business measure is represented by a separate column in one of the two tables. Depending on the options selected during the building of the business measures models, much or all of the information in the State database is replicated to the Runtime database.

The State database is used by Monitor server. At runtime, the Monitor server inserts, retrieves, and updates the information of processes instances that reside in the State database, according to the processed events.

The State database stores the following information:

- Information about business measures groups, which is a part of the data in the imported business measures models.
- The running process instances that are created while the Monitor is running.

- The event entries of the running processes. The event entry is the event data that is received for updating a specific business measures group.

Runtime database

The Runtime database is similar in structure to the State database. It receives replicated information from the State database about the current status of all running processes as well as the final status of recently completed or failed processes. This information is used by Monitor dashboards. The Runtime database is also used by the Adaptive Action Manager to store alert notifications. There is only one Runtime database per Monitor installation.

The information in the Runtime database is replicated from the State database. The Runtime database stores:

- Alert notifications sent by the Adaptive Action Manager to the dashboards
- Process instance data
- Metrics values

The Runtime database is used by the Monitor dashboards. The dashboards retrieve the running or recently completed instances data required to populate the views from the Runtime database. The dashboard views use the Runtime database for analytical purposes, so it is optimized for query processing and aggregate query processing.

All completed instances remain in the Runtime database for 24 hours and are deleted afterwards. 24 hours is the default retention policy which can be modified as part of the data movement service configuration.

History database

The History database stores all completed and running process instances. It is used by the dashboards for enhanced data analysis using DB2 Alphablox. There is only one History database per Monitor installation. The data in the History database is never deleted.

The History database should only contain two years worth of historical data. This is one of Monitor product requirements. As mentioned before, the historical data is never deleted automatically, so the DBA is responsible for deleting the data that is greater than two years old.

The History database stores the information regarding long-running instances as well as completed instances. This information is stored as star schemas rather than in the flat transactional forms used in the State and Runtime databases. The History database is optimized for aggregated and long running queries. It is used by DB2 Alphablox in dashboards views to provide advanced multidimensional reports.

The information in the History database is replicated from the Runtime database.

The History database contains dynamic tables that are created according to the deployed business measures model. The schema generator generates the History database schema, which is used to create dynamic tables, and Cube Views definitions.

The History database is used by the Monitor dashboards. The dashboards retrieve the data required to populate some views from the History database. For example, the Reports view focuses on analyzing data extracted from the History database.

4.4.6 The Monitor dashboards

Business Monitor has WebSphere Portal-based dashboards that can be customized with a predefined set of configurable views that are implemented via portlets. A dashboard designer with WebSphere Portal skills can take these portlets and configure them or create custom dashboards. Some of these view types will leverage DB2 Alphablox for a more refined data analysis.

Measuring the process using the Dashboard Client component of WebSphere Business Monitor Version 6.0 enables users to monitor business performance through a set of views. The following are examples of those views, that can be combined to create dashboards:

Scorecard view

A scorecard is a set of performance measures that are linked to objectives and goals of a specific business area. Business users select the KPIs pertaining to their various areas of responsibility and place them in perspectives (categories) on their scorecards. On the Scorecards view, users can easily watch the runtime values of vital numeric KPIs, monitoring them against their defined targets. This is depicted in Figure 4-51.

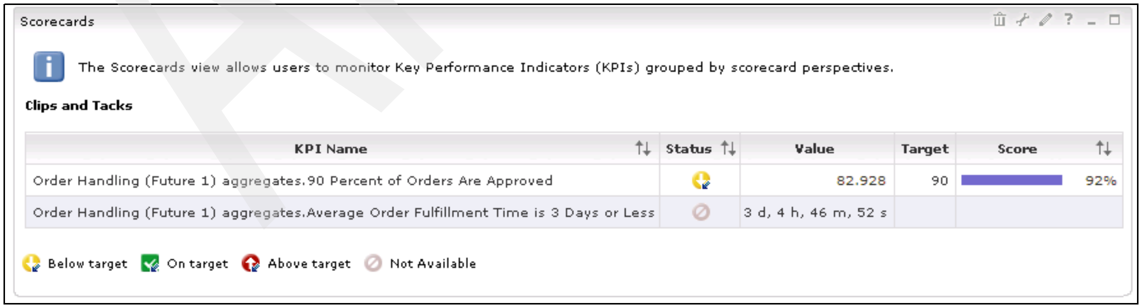


Figure 4-51 Scorecard View

KPI view

This view shows the values of individual KPIs, focusing on the position of the KPI value relative to its defined limits. It displays all KPI information so that business users can easily monitor them and take action if necessary, as depicted in Figure 4-52. For this, you must have a business measures model that contains the KPIs to be monitored, with their threshold limits.

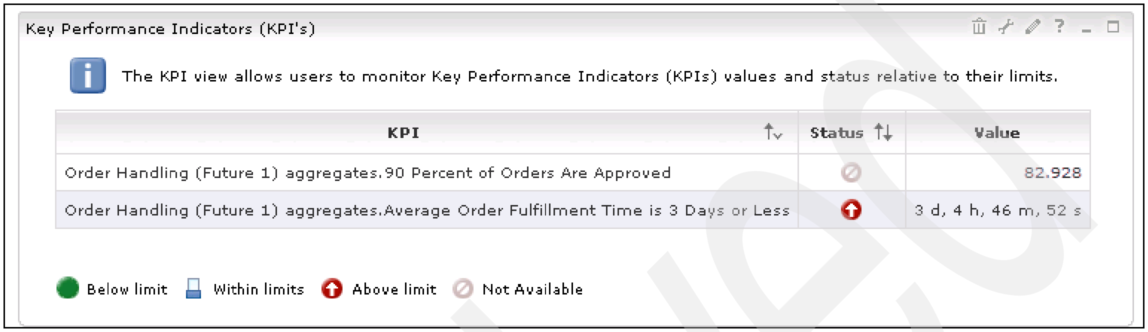


Figure 4-52 KPI View

Gauge view

This view displays individual KPI values, either relative to KPI threshold value or relative to the KPI target value. It visually uses a paradigm of physical instruments in the form of gauges and dials, such as automobile speedometers or temperature gauges. This is depicted in Figure 4-53.

Gauges help you to visualize information by representing KPI values. They have a focus on representing numeric KPIs that belong to aggregated business measures in a business measures model. Each gauge represents the value of a single KPI.

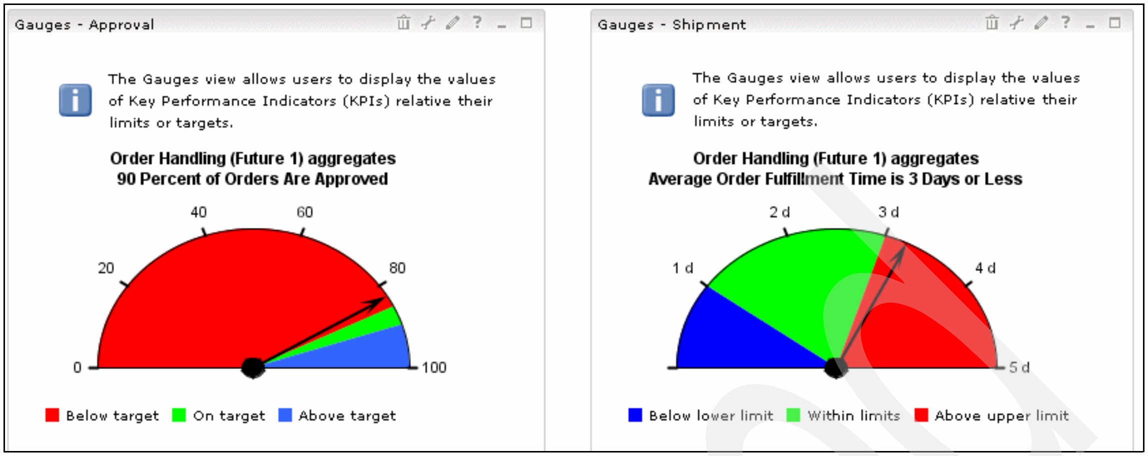


Figure 4-53 Gauge View

Active Instance view

This view shows the details of processes, which contain a group of related metrics, KPIs and situations in a particular process. The monitor view can monitor either individual instances or groups of process instances. This is in addition to providing the capabilities to perform drill-down and drill-up among child and parent instances.

The Active Instances view shows the details of a process at run time, and displays information about running instances. You can monitor values of KPIs and metrics that belong to an aggregate business measures group, along with stopwatches and counters. You can also drill down to view the underlying activities, such as items in process instances, and whether they are realized by activities, local subprocesses, or global subprocesses.

The Active Instances portlet is shown in Figure 4-54, and consists of a Active Instances table that displays the business measures of each currently running process instance.

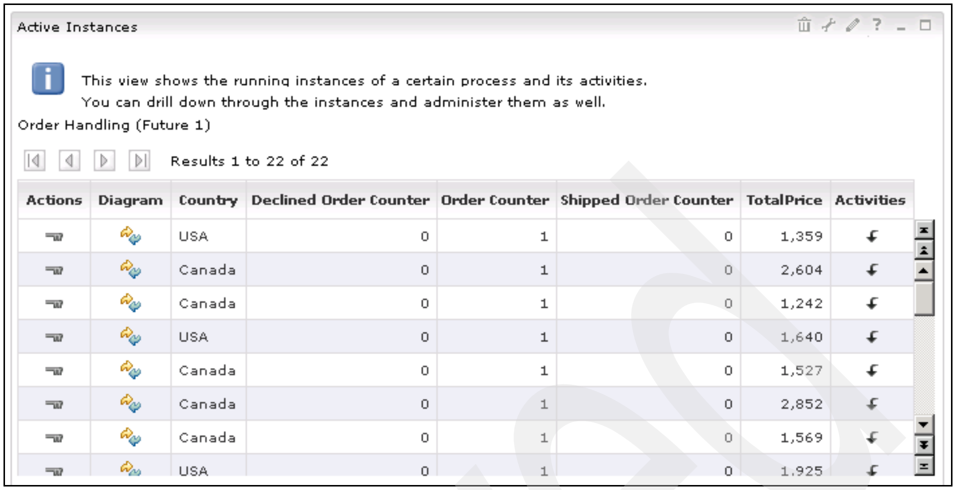


Figure 4-54 Active instances

Other views are:

- Alert view displays the notifications for a specific user.
- Report view provides performance reports relative to a time-axis. Such reports contain tables and graphs for analyzing historical data contained within the performance warehouse data store. The Report View has built-in analysis types that include Quartile, Trend, and Control Analysis.
- Dimensional view provides a multidimensional view of business performance data. Users can pivot on business dimensions to view the performance.
- Process view displays the process status. Such a display will show a process graph with visual cues on the status of the process.

In addition to the previous views, there are some helper views such as:

- Organizations view: Displays the organization, organization units and employees available in the customer user registry that WebSphere Portal is configured to use, whether that is an LDAP user registry or a custom registry. The Organization View is used to help users filter the data generated by the Report View based on employees available in selected organization or organization unit, or based on a selected employee.
- Export values view: The purpose of this view is the export of an XML representation of the actual values to be used for the feedback loop with the WebSphere Business Modeler.

A launchpad installation helps you effectively deploy WebSphere Business Monitor, which includes a limited license of specific components:

- DB2 UDB Enterprise Server Edition
- DB2 Cube Views
- DB2 Alphablox
- WebSphere Portal
- WebSphere Process Server

4.4.7 WebSphere Process Server and Integration Developer

Based on SOA and as a single, simplified programming model, WebSphere Process Server V6.0 is a business process server that delivers and supports all styles of integration based on open standards to automate business processes that span people, workflows, applications, systems, platforms, and architectures.

WebSphere Integration Developer V6.0 is based on Eclipse technology, and is a new tool for rapid assembly of business solutions that allows you to describe all styles of processes with one programming model based on Business Process Execution Language (BPEL).

- Service component architecture: One simplified integration framework that leverages existing IT.
- One tool to describe all processes: Visual editors for component development, assembly, integrated testing, and deployment.
- Support for all styles of integration: Including human tasks, roles based task assignments, and multilevel escalation.Visual editors for component assembly.
- Change business processes dynamically.
- Business rules, business state machines, and selectors to dynamically choose interface based on business scenarios.
- Broadest reach in integration: Built on Enterprise Service Bus (ESB) technologies and support for IBM WebSphere Adapters.
- Support for business-to-business (B2B) through a restricted use license of IBM WebSphere Partner Gateway.

IBM WebSphere Integration Developer V6.0 and IBM WebSphere Process Server V6.0 deliver a composite application platform optimized for building service-oriented applications that extend and integrate a company's existing IT assets.

Note: Some WebSphere product names have changed. As examples, WebSphere Process Server V6.0 is the successor to WebSphere Business Integration Server Foundation 5.1.1, WebSphere Integration Developer is the successor to WebSphere Studio Application Developer Integration Edition V5.1.1, and WebSphere Process Server V6.0 is the successor to WebSphere Business Integration Server V4.3.

You should be aware that over time, a few product names have been changed. As examples, WebSphere Process Server V6.0 is the follow-on product to WebSphere Business Integration Server Foundation 5.1.1 and WebSphere Integration Developer is the follow-on product to WebSphere Studio Application Developer Integration Edition V5.1.1. WebSphere Process Server V6.0 is also the follow-on product to WebSphere Business Integration Server V4.3.

WebSphere Process Server V6.0 offers process automation, advanced human workflow, business rules, Application to Application (A2A), and B2B capabilities all on a common, integrated SOA platform with native Java Message Service (JMS) support.

WebSphere Process Server builds on the WebSphere Application Server to provide a Java 2 Enterprise Edition (J2EE) and Web services technology-based application platform for deploying enterprise Web services solutions for dynamic on demand business.

WebSphere Process Server includes all of the features available in WebSphere Application Server Network Deployment V6 including J2EE 1.4 support, Web Services Gateway, IBM Tivoli Performance Viewer, clustering and workload management support.

Included in these products are several complementary products for use in the WebSphere Process Server environment including restricted use licenses for DB2 Universal Database (UDB) Enterprise Edition, Tivoli Directory Server, and Tivoli Access Manager.

WebSphere Process Server also includes a restricted use license of WebSphere Partner Gateway Advanced Edition to provide consolidated partner services for process integration with the WebSphere software platform. Business-to-business (B2B) gateway consolidation centralizes B2B communications with trading partner communities, providing a central point of control for interactions among partners, and providing a security-rich environment at the edge of the enterprise. B2B gateway consolidation is of particular value when multiple business units interact with the same partners or partners with similar processes.

WebSphere Partner Gateway combines extensive partner profile management capabilities with a simple, reliable and secure exchange for B2B messages capable of serving multiple B2B protocols and standards to efficiently integrate your processes with those of your business partner community.

WebSphere Integration Developer V6, optimized for developing composite applications that deploy to WebSphere Process Server V6, delivers an application development environment for building service-oriented, component-based applications that extend and integrate your IT assets.

WebSphere Integration Developer V6.0 focuses on developer productivity by providing authoring tools allowing integration developers to build and debug composite business integration applications. Combined with other development tools from IBM, for example, Rational Application Developer and WebSphere Business Modeler, it provides support for enterprise developers.

Process server and integration developer - together

Together WebSphere Process Server V6.0 and WebSphere Integration Developer V6.0 provide a comprehensive list of services to enable the development of composite integration applications. These service components include:

- Business processes
- Human tasks
- Business state machines
- Business rules
- Supporting components

Business process

The business process component in WebSphere Process Server implements a WS-BPEL compliant process engine. It represents the fourth release of a business process choreography engine on top of the WebSphere Application Server.

WS-BPEL defines a model and a grammar for describing the behavior of a business process based on interactions between the process and its partners. Support for WS-BPEL includes:

- A business process editor with authoring experience.
- Drag-and-drop tools to visually define the sequence and flow of WS-BPEL business processes.
- A visual business process debugger to step through and debug WS-BPEL business processes.
- Long- and short-running business processes.

- Compensation support to provide transaction rollback-like support for loosely coupled business processes that cannot be undone automatically by the application server.
- Integrated fault handling to provide an easy and integrated means of performing in-flow exception handling.
- Support for including Java snippets and artifacts as part of a business process.

Human tasks

Human task support expands the reach of WS-BPEL to include activities requiring human interaction as steps in an automated business process. Business processes involving human interaction are interruptible and persistent (a person may take a long time to complete the task) and resume when the person completes the task.

Human task support includes:

- Staff activity nodes to represent a step in a business process that is performed manually.
- Ability to assign people (for example, direct reports) to specific instances of a process via staff queries that are resolved at runtime using an existing enterprise directory.
- Graphical browser-based interface for querying, claiming, working with, completing, and transferring work items to another user.
- Java Server Faces (JSF) components to create custom clients.
- Advanced work item management support for managing the creation, transfer, and deletion of work items.
- Dynamic setting of duration and calendar attributes for staff activities.
- Dynamic setting of staff assignment via custom attributes.
- Originating task support to invoke any kind of service (including a business process).
- Ad-hoc creation and tracking of human tasks in addition to the "claim, save, transfer and complete" model for processing and tracking of tasks which has been available in since WebSphere Business Integration Server Foundation 5.1.
- Administrative tasks.

Human tasks can be used to invoke services (for example a business process), participate in a business process (traditional Staff Activity) or administer a business process (process administrator). Additionally pure Human Tasks are available to implement Ad-hoc processing. By separating human task support from the core WS-BPEL engine WebSphere Process Server and WebSphere Integration Developer now allow creation of pure WS-BPEL code without IBM extensions for human tasks.

Business state machine

WebSphere Process Server V6.0 provides a business state machine component that can be used to model heavily event-driven business process scenarios. These kind of event-oriented scenarios are sometimes hard to model in a WS-BPEL model, but very easy to model in a state machine diagram. This state machine is modeled after the Unified Modelling Language (UML) state machine diagrams. The combination of WS-BPEL business processes with business state machines gives WebSphere Process Server V6.0 a unique edge when it comes to business process automation.

Business rules

WebSphere Process Server V6.0 contains a business rule component that provides support for Rule Sets (If/Then rules) and decision tables. Business rules are grouped into a Rule Group and accessed just like any other component.

WebSphere Process Server V6.0 also provides a Web client with national-language-supported plain text display capabilities to allow on-the-fly changes to business rules to be deployed using an intuitive user interface. By separating the business rules component from the individual business process flows, a rule can be managed by the domain expert for that particular business rule and by encapsulating rules as a service component a rule can be used across multiple processes for maximum business flexibility.

Supporting components

WebSphere Process Server V6.0 provides a wide range of supporting components in order to facilitate component-based application development. Among these are:

- Interface maps: Can be used to convert semantically but not syntactically identical interfaces. These are very beneficial for importing existing services which may have a different interface definition than required. They are also beneficial for implementing a completely canonical integration solution where one component has no knowledge of the implementation details of another component.
- Data maps: Can be used to translate one business object into another — for example as part of an interface map it is often necessary to translate the arguments of an operation.
- Relationships: Can be used to convert key information to access the same data sets in various back-end systems and keep track of which data sets represent identical data. This enables cross-referencing and federation of

heterogeneous business objects across disparate Enterprise Information System (EIS) systems. Relationships can be called from a business object map when converting one business object into another to manage the key information. Additionally lookup relationships can be defined for static data (for example, mapping zip codes into city names).

- Selectors: Can be used to dynamically invoke different components based on various rules (for example, date). Combined with Interface Maps a great deal of flexibility can be achieved. A Web Interface is provided to change these selector rules on the fly for example to invoke a newly deployed module without having to redeploy the calling module.

Back-end system connectivity

WebSphere Integration Developer V6.0 provides integrated, open standards-based support for building composite applications including WS-BPEL business processes that integrate with back-end systems including:

- Integrated tool support for using J2EE Connector Architecture (J2C) 1.0 and 1.5 resource adapters to access back-end systems.
- Tool integration for J2C adapters with tool plug-in extensions (available from IBM and IBM Business Partners).
- J2C 1.5 resource adapter support to leverage WebSphere Adapters.
- Support for the entire suite of WebSphere Business Integration Adapters.
- Tools for creating services out of J2C resource adapters or WebSphere Business Integration Adapters and including those services as part of an integration application.
- Wizards to manage the low-level data handling requirements for J2C resource adapters.
- Support for Web services (JSR 109 / JAX-RPC-based).
- Support for JMS messaging through the integrated WebSphere messaging resources (with full connectivity to existing WebSphere MQ based networks).
- Support for calling EJB™ Session Beans.
- Wizards to quickly and simply expose CICS® or IMS programs as enterprise services including the ability to import definitions from COBOL, C structures, CICS basic mapping support (BMS), and IMS Message Format Service (MFS) definitions.

WebSphere Process Server V6.0 builds on the WebSphere Application Server to provide a J2EE and Web services technology-based application platform for deploying enterprise Web services solutions for dynamic e-business on demand.

4.5 Application Connectivity

Application Connectivity integrates people, partners, processes and applications on a common infrastructure, for maximum efficiency. You can address your integration needs, regardless of where you are in the spectrum or continuum of integration, from simple connectivity all the way to the requirements of a fully integrated, on demand enterprise.

EAI is designed to address the following needs:

- Application integration is the engine that drives business end-to-end.
- Integrate applications on a common infrastructure, for maximum efficiency.
- Address your integration needs from simple connectivity all the way to the requirements of a fully integrated, on demand enterprise.
- React rapidly to business events, reduce inventories, shorten new process cycle times and lower costs by reusing existing process and applications and other IT assets.
- Drive horizontal processes across vertical applications and across systems.
- Improve service and business agility by responding to business driven integration changes rapidly and at lower cost. Better position your company for mergers and acquisitions.
- Optimize dealings with partners and customers with a single view of data.
- Easily add new packaged applications and systems without disrupting existing business processes.
- Improve tooling using the Eclipse framework which means cycle time from LOB idea to IT production are greatly minimized and do not need Java developers.

The products that comprise an EAI solution are:

- WebSphere MQ: Enables application integration by helping business applications exchange information across different platforms, sending and receiving data as messages. For further information refer to “WebSphere MQ” on page 217.
- WebSphere Message Broker: Delivers an advanced Enterprise Service Bus for providing connectivity and universal data transformation for both standards- and non-standards-based applications and services. In particular, it provides Web services connectivity, JMS messaging, and service oriented integration to power your service oriented architecture (SOA). There is also WebSphere Enterprise Service Bus, which is more typically used for a predominantly Web services-based environment.

- WebSphere Partner Gateway: Enables your business to extend integration beyond your enterprise, integrating with trading partners to create a community of businesses, exchanging information and sharing processes.

4.5.1 WebSphere Message Broker

WebSphere Message Broker integrates different applications and systems, providing transformation and enrichment of in-flight information to provide a level of intermediation between applications that use different message and data structures and formats. The product enables applications to work together so that they exchange information as though they had been designed to do so from the start. It provides the ability to have a range of connectivity options between applications, to meet both the needs of the applications and the requirement for the distribution of the integrated data, and provides all this within a comprehensive environment separate from the application development. Thus, users such as application developers can concentrate on business logic without reducing application and business flexibility by custom coding connectivity and integration logic in the applications and services throughout the business.

WebSphere Message Broker enriches and distributes real-time information from virtually any source of information held in any format through a network of access points or a centralized broker and out to multiple receiving endpoints, each provided with tailored data, and this can provide a powerful approach in unifying the IT potential of an organization.

Connectivity to and from WebSphere Message Broker can take advantage of the assured delivery offered by products such as WebSphere MQ or WebSphere Application Server, meaning that transactions will take place and complete even after a temporary network failure, so that users and their customers can be confident that information, which is transmitted as messages, will be delivered.

With WebSphere Message Broker, users can connect and integrate nearly all types of applications and data to almost any endpoint through a single extensible and configurable environment. The graphical Eclipse-based programming environment of WebSphere Message Broker provides users with a powerful and architected integration layer which will help them to avoid the burden of writing complex bespoke programming as a part of each application or to make use of each data source. Because WebSphere Message Broker provides functions such as reusable components, adding new connections, applications, or interactions with data is vastly simplified and releases programmers to concentrate on new and changing business requirements.

Connectivity and information sharing between applications for environments that were never designed to work together free users from manual re-keying of data from one application to another. All of an organization's business data is accessible in real-time in usable formats to every part of the business. Users can access data faster and respond better to customer needs. Changes to the business can be implemented faster, with fewer errors, and with no disruption to the business applications.

With WebSphere Message Broker, customers can extract valuable data from existing applications and other sources of data and use it elsewhere in their business. They can even access and widely distribute data from applications for which they no longer have the source code. Users do not need to make costly changes to the applications to take a different view of the data they provide.

Users can be more confident that the information they share with their customers is accurate, timely, and (through taking advantage of the reliability and recoverability offered by WebSphere Message Broker) enhanced when used to connect systems using the assured delivery features of WebSphere MQ.

4.5.2 Information distribution

In a service-oriented architecture (SOA), the enterprise service bus (ESB) optimizes information distribution between service requesters and service providers. Each enterprise can deploy its own unique ESB, reflecting how far it has advanced toward becoming an on demand business.

As key members of the IBM WebSphere software portfolio, WebSphere MQ and WebSphere Message Broker enabled you to begin deploying or widening the deployment of your ESB. There is also WebSphere Enterprise Service Bus, for predominantly Web services-based environments.

Some businesses may find that simple messaging-based connectivity between well-matched applications provides the aspects of integration that they require to implement an effective enterprise service bus. Other businesses may find that to extend these capabilities into their wider deployed infrastructure, they realize value of other parts of the WebSphere software portfolio. As key members of the IBM WebSphere software portfolio, WebSphere MQ and WebSphere Message Broke, and WebSphere Enterprise Service Bus, enable deployment of an ESB.

These programs help maximize the value of your IT investment by broadening the range of environments this connectivity layer can reach, such as hardware and operating system platforms and non-standards-based programming models as well as J2EE and .NET.

For connecting these diverse programming models there needs to be proven delivery of messages and data between the applications. The proven delivery mechanisms of WebSphere MQ and WebSphere Application Server can extend the standards-based enterprise service bus with reliable connectivity throughout the enterprise. WebSphere Application Server messaging resources provide a best-of-breed Java Message Service (JMS) implementation for use with J2EE-hosted applications. WebSphere MQ seamlessly extends those messaging resources to non-J2EE environments to integrate virtually anything, across more than 80 platforms.

WebSphere Message Broker adds services such as message routing, transformation enrichment, and support for a range of message distribution options and protocols to improve their flexibility and performance. This enables businesses to integrate virtually any applications on any systems, exchanging their data together in real-time.

WebSphere software provides integration capabilities that enable reaping of the benefits of service-oriented architectures with connectivity and integration through the WebSphere Enterprise Service Bus.

4.5.3 WebSphere MQ

WebSphere MQ provides reliable, resilient application integration by passing messages between applications and Web Services. It reduces the risk of information loss and the need to reconcile communicating IT systems by using queuing and transactional facilities that help preserve the integrity of message across the network. It also helps to eliminate the need to write complex communications code due to the APIs (Message Queue Interface (MQI) and JMS) already written.

WebSphere MQ simplifies integration tasks by providing a function application development interface that removes the data movement and storage logic from application.

In the real-time enterprise environment, it is a base product. It is used to create Q Replications, and host event publishing and WebSphere Business Integration flows to close the real-time enterprise loop.

4.6 Balanced Configuration Unit

Right-time BI systems require a configuration of robust components, including analytical and data warehousing software, servers, storage and databases. Improper configuration designs can be costly in terms of time and money when implemented on too many servers, with too much storage, and with inadequate software. To help with appropriate sizing, IBM offers the balanced configuration unit (BCU).

The BCU originated from the IBM Business Intelligence Best Practices Team, a group that has years of extensive experience in designing and configuring data warehouse solutions. Using their knowledge and expertise, IBM has created a simplified solution designed for customers to properly size, build, and grow their data warehouse solution with more predictability and less risk, while reducing their time to market.

The concept of modular scaling is a significant advantage. Not only does it allow the BCU to maintain balance with the initial build and installation, but the BCU design provides a predictable and scalable plan for growth as both planned and unplanned data warehouse workload demands increase over time.

By mapping your business requirements to a pool of resources (nodes), when your workload expands your simply add additional blocks to scale. Selecting the BCU gives you the security of knowing that your existing infrastructure is based on best practices principles, and that you have a prescriptive(?), scalable approach to grow the data warehouse while maintaining total system balance that is optimized for its size and workload.

Utilizing the power of the shared nothing architecture of DB2, coupled with the business intelligence (BI) tools of Data Warehouse Edition (DWE), you can create stand-alone building blocks that can be scaled to suit the needs and size of your data warehouse.

The BCU building blocks are constructed of server, storage, and operating system components that have been chosen for optimal performance, and balance each other's characteristics. Balance is a key factor when designing a warehouse.

The BCU for data warehousing is a proven approach for building successful data warehouse infrastructures. The BCU was designed to deliver major benefits to customers building large scale data warehouses. The BCU implementation reduces the time required for Business Intelligence infrastructures to get up and running, The BCU takes advantage of Industry-standard components for ease of installation and implementation, balanced performance for delivering scalability and fault tolerance for high availability. Therefore, providing faster time to market, lower total cost of ownership and high reliability.

The BCU provides the following three key advantages:

1. IBM optimized components.
2. Repeatable, scalable, consistent performance that can grow as business needs grow.
3. Prescriptive, validated and tested, best practices design that reduces the time, and removes the risk of building a business intelligence solution.

A Balanced Configuration Unit (BCU) is composed of software and hardware that IBM has integrated and tested as a pre-configured building block for data warehousing systems. A single BCU contains a balanced amount of disk, processing power and memory to optimize cost-effectiveness and throughput. It is basically used to reduce design time, shorten deployments, and maintain strong performance ratios as they add building blocks to enlarge BI systems.

The Balanced Configuration Unit (BCU) is a complete solution for building a BI infrastructure using a modular building block design with a focus on data warehousing.

The BCU definition consists of:

Unit = Building block or logical entity, This allows ease of expansion for growth.

Configuration = Design specification using preselected a software package and hardware platform.

Balanced = Meaning a scalable performance ratio of disk, I/O, memory, CPU and Network for predicable performance.

4.6.1 Balanced Partition Units

The building blocks of the BCU start with a virtual entity, the BPU (Balanced Partition Unit). The BPU is one DB2 partition and its related components. Components refer to the processes and hardware resources that service each individual partition (BPU) and consist of DB2 agents, memory, and disk. The BPU is built on specifications that have been developed over time based on extensive experience and is significantly optimized for data warehousing workloads. For BCU design purposes the BPU is considered the smallest element used in sizing a configuration. A BPU is depicted in Figure 4-55.

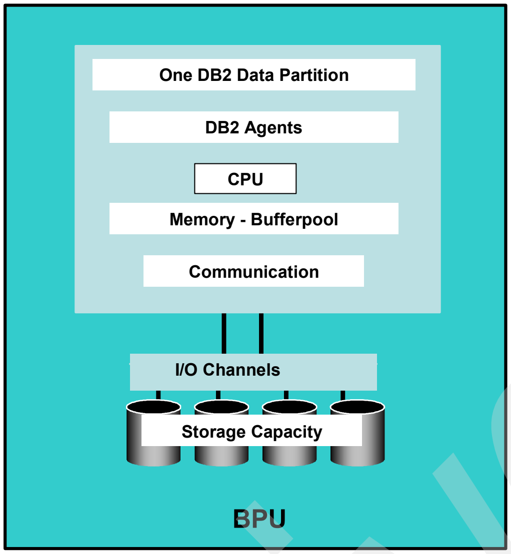


Figure 4-55 BPU building block

There are five types of Balanced Partition Units and each one configured uniquely to address a specific purpose.

- Data BPU: a DB2 database partition dedicated to managing both non-partitioned table data and multi-partitioned table data within a DPF environment. In a DB2 DPF environment, there are two basic kinds of data tables: tables which are partitioned across multiple database partitions, and tables (typically smaller or more static data sets) that are maintained in a single tablespace.
- Coordinator BPU: manages user connections and coordinates queries.
- Catalog BPU: where the DB2 catalog resides.
- ETL BPU: Using the Data Warehousing Balanced Configuration Unit design allows you to achieve a scalable system to support future growth with minimal effort. An ETL BPU consists of a collection of virtual resources that serve the functions of a conceptual partition where the ETL processing occurs. It has all the resources that the ETL requires.
- Application BPU: a conceptual partition where the application processing typically occurs, and which has all the resources that the application requires. An example of an application BPU would be the WebSphere Application Server running DB2 Alphablox.

Although the catalog, the coordinator, and single partition tables share a single partition, you need to assess each of these functions separately for sizing the system. And, as the data warehouse grows, especially in terms of concurrent usage, you may find that there is benefit in adding one or more additional coordinator partitions (nodes). These database functions have been separated because each handles a different type of work and each may require different resources or possibly different configurations. So, it is quite useful to consider these functions independently. Each of these functions could also be supported by the resources of a DB2 database partition (BPU) dedicated only to the specific function. Thus the terms Catalog BPU, Coordinator BPU, and Single Partition Tables BPU". However, as a default best practice all these functions should reside within one DB2 database partition and can be referred to as the Administration BPU. As a result, the administration BPU would generally be on a separate server and sized differently than the Data BPUs. The administration server is where Query Patroller (QP) and the QP control tables would be installed to manage your query workload. QP provides a high value query governing function, so consideration of QP workloads need to be taken into account when sizing your system.

5

More real-time enterprise enablers

If we come to the conclusion that we cannot currently achieve the status of having a real-time enterprise, what can we do? The answer might be found in the story concerning an academic and an engineer. The academic comments that if we start walking toward a wall, but each step is only half the size of the previous, we will never reach the wall. The engineer replies that it does not matter since we will get close enough for all practical purposes.

The question is then, can we get close enough to a real-time enterprise for all practical purposes? The answer in most cases, is most definitely! To get there requires a real-time data warehousing environment. Or, those who like to be a bit more specific might call it near real-time data warehousing. Or perhaps near real-time business intelligence. In fact, many have been heading down that path for some time. And one of the steps along the way was to build something called an Operational Data Store (ODS).

The ODS is typically updated in something close to real-time. However, the ODS is a reflection of the operational systems. It is built primarily for operational purposes, and is typically not optimized for analytical applications. Never the less, the ODS is a component of the data warehousing environment, and as such is used with the data warehouse to provide business intelligence that is closer to real-time. For more information about this ODS topic, refer to the redbook

Building the Operational Data Store on DB2 UDB Using IBM Data Replication, WebSphere MQ Family, and DB2 Warehouse Manager, SG24-6513.

Real-time data warehousing typically implies some type of asynchronous feeding of the data warehouse from the source systems that will involve some type of latency typically greater than zero - with zero being the optimum for real-time. However, asynchronous data feeds and zero latency are not requirements to be considered a real-time enterprise. Here we start to get into semantics, rather than taking the practical approach - which is to improve the efficiency and effectiveness of your business intelligence.

As we look at moving forward in pursuit of a real-time data warehousing process, we have to consider how we:

- Capture the data.
- Deliver that data to the data warehouse environment.
- Apply transformations to the data.
- Update the data warehouse while concurrently allowing access to it by queries and applications.

We also have to consider the need to further feed the data to downstream aggregates or data marts.

We have defined four recommended update approaches:

- Continuous flow with WebSphere Information Integrator
- Continuous flow with WebSphere MQ and DB2 Listener
- Using SQL generators and applications (extract, load, and transform (ELT) rather than extract, transform, and load (ETL))
- Continuous flow with an ETL tool such as WebSphere DataStage

In the following sections, we look at these update approaches.

5.1 Continuous flow processing

Replication is one of the primary technologies used to update and maintain the data warehouse, and enable access to data that is more current. For example, with replication you can capture database changes from transaction databases and replicate them into operational data stores, data warehouses, or data marts to facilitate real-time business intelligence.

There are two different solutions that you can use to replicate data from and to relational databases: SQL replication and Q replication. In SQL replication, committed source changes are staged in relational tables before being replicated to target systems. In Q replication, committed source changes are written in messages that are transported through WebSphere MQ message queues to the target systems.

There is also a solution called event publishing for converting committed source changes into messages in an XML format and publishing those messages to applications such as message brokers.

5.1.1 SQL replication

SQL replication captures changes to sources and uses staging tables to store the committed transactional data. The changes are then read from the staging tables and replicated to corresponding target tables. With staging tables, data can be captured and staged once for delivery to multiple targets, in different formats, and at different delivery intervals.

Replication can be continuous, at intervals, or for one time only. Replicating continuously can be useful if your enterprise applications need data in near real-time. Replicating at intervals can be useful for replicating large batches of data during off-peak hours. You can also trigger replication through database events.

There is considerable flexibility with the data that you want to replicate. For example, you can choose to have all rows and columns replicated or you can choose just a subset. By subsetting rows and columns, you transport across the network only the data that you want. You can also clean, aggregate, or otherwise manipulate the data. If you manipulate data, you can do so at the source and then distribute the manipulated data. Or you can manipulate the data when you replicate it so that some targets get manipulated data and others do not.

The SQL replication process is depicted in Figure 5-1. The capture program reads the committed transaction data from the source DB2 log tables. That data is then available to the apply program, that writes the data updates to the target tables. Control tables are used to synchronize the activities of both the capture and apply programs to ensure data integrity.

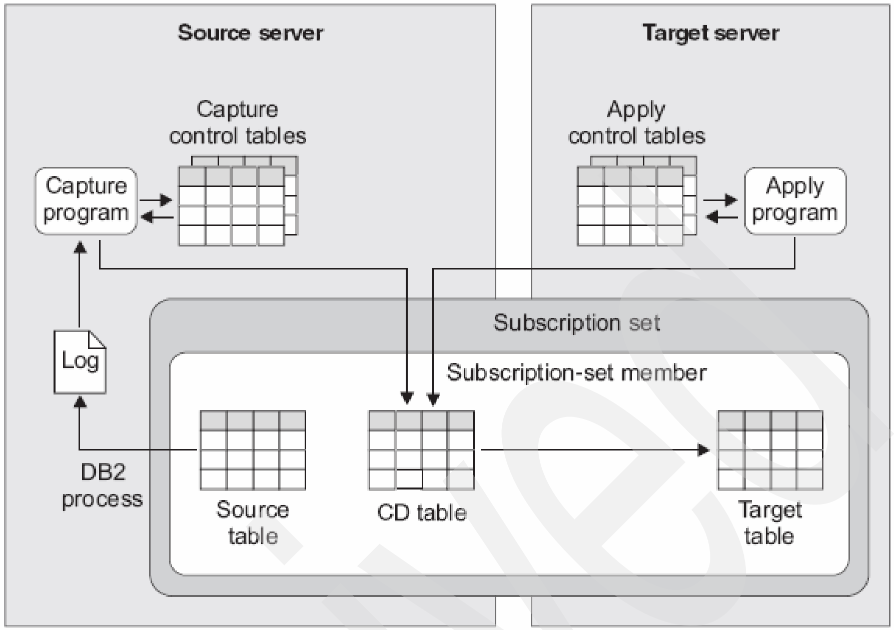


Figure 5-1 SQL Replication

5.1.2 Q-replication

Q replication is a solution that can replicate large volumes of data at very low levels of latency. Q replication captures changes to source tables and converts committed transactional data to messages. This data is put into queues as soon as it is committed at the source, and then read by Q replication. This technology is different from SQL replication. For example, the data is not staged in tables. In this process, the data is put in messages that are sent to the target location through WebSphere MQ message queues. Here, the messages are read from the queues and then converted back into transactional data. The transactions are then applied to the target tables with a highly parallel method that is very fast, and still preserves the integrity of the data.

You can replicate a subset of columns and rows from the source tables. All subsetting occurs at the source location so that only the data you want is transported across the network. If you want to perform data transformations, you can pass replicated data to your own stored procedures.

This process is depicted in Figure 5-2. Here the committed transaction data is read from the DB2 log tables by the capture program. However, the data is then put into WebSphere message queues rather than into DB2 tables. The apply program is activated when there is a message in the queue. It reads the message and applies the changed data to the DB2 target table. As with SQL replication, there are control tables to synchronize the activities of both the capture and apply programs to ensure data integrity. WebSphere MQ guarantees delivery of the data messages, and thus data integrity.

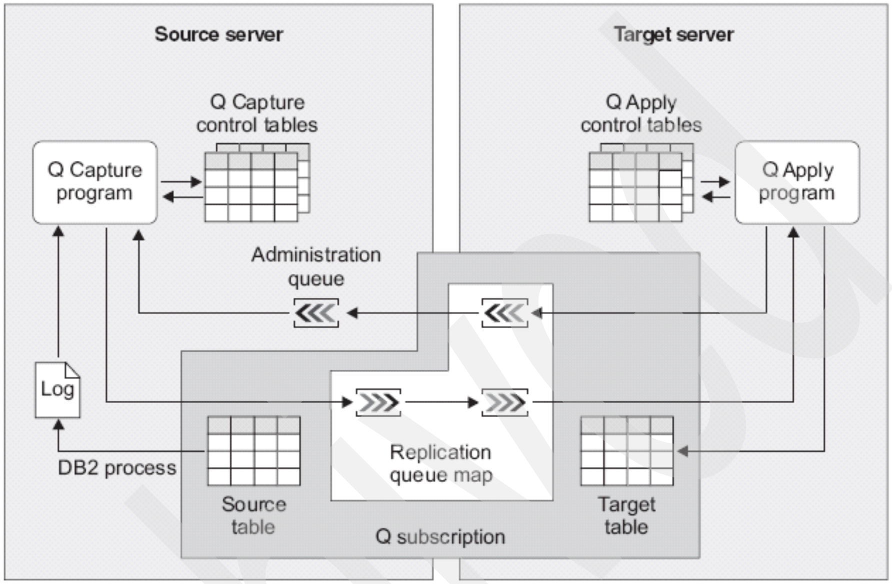


Figure 5-2 Q Replication

5.1.3 Event publishing

The event publishing solution captures changes to source tables by reading from the DB2 log tables. That committed transaction data is then converted to messages, much like Q replication. However, with this technique, the data is translated into the Extensible Markup Language (XML) format. Each message can contain an entire transaction, or only a row-level change. These messages are put on WebSphere MQ message queues and read by a message broker, or other applications. You can publish subsets of columns and rows from source tables so that you publish only the data that you need. The data is subsequently read from the message queue by the consuming application.

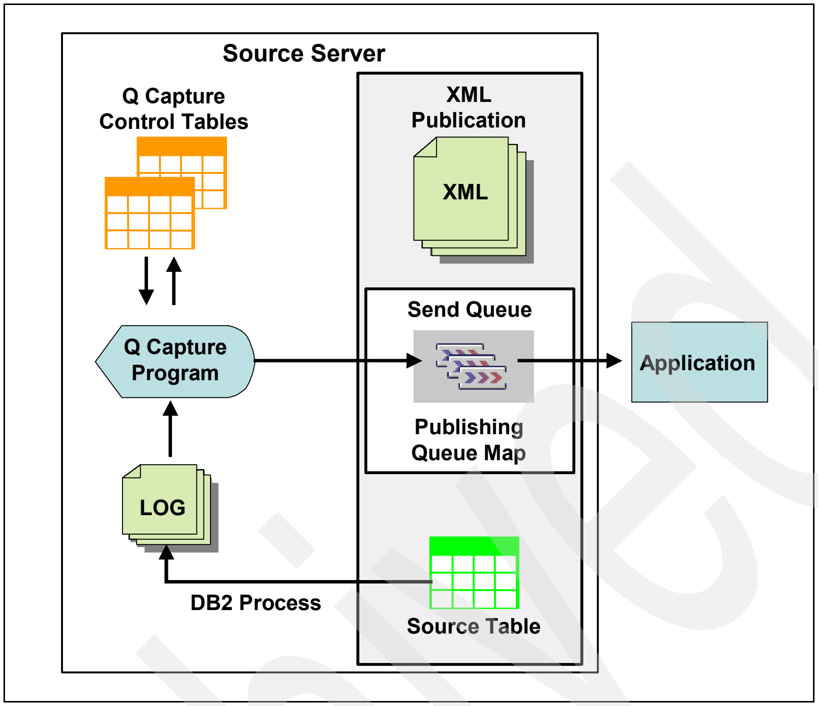


Figure 5-3 Event Publishing

5.1.4 Comparing the technologies

In this section we offer a comparison table for SQL replication, Q replication, and event publishing. For ease of comparison it is presented in table format, and based on a number of relevant attributes for replication. The comparison is depicted in Table 5-1.

Table 5-1 SQL Replication, Q Replication and Event Publishing

Points of Comparison	**SQL Replication**	**Q Replication**	**Event Publishing**
Uses	Multiple, including capacity relief, feeding data warehouses and data marts, and auditing change history.	Multiple, including failover, capacity relief, supporting geographically distributed applications, data availability for planned or rolling upgrades or outages.	Multiple, including feeding central information brokers and Web applications, and triggering actions based on updates, inserts, or deletes to source tables.
Data Replication	Committed transactional data is captured and stored in staging tables by a capture program. An apply program reads the information from the staging tables and applies the transactions to target tables. Committed transactional data is captured and stored in staging tables by a capture program. An apply program reads the data from the staging tables and applies the transactions to target tables.	Committed transactional data is captured and put on WebSphere MQ message queues by a capture program. An apply program reads the data from message queues and applies the transactions to target tables.	Committed transactional data is captured and put on WebSphere MQ message queues by a capture program.
Sources	Tables (in DB2 UDB or other supported relational database management systems), or views.	Tables in DB2 UDB	Tables in DB2 UDB

Points of Comparison	SQL Replication	Q Replication	Event Publishing
Targets	Tables (either in DB2 UDB or other supported relational database management systems), or views	Tables in DB2 UDB	Tables in DB2 UDB
Filtering	Column and row filtering	Column and row filtering	Column and row filtering
Where the filtering is done?	Source, Target or both	Source	Source
Data Transformation	Data cleansing, data aggregation, and calculated columns in target tables. Manipulate data once and replicate to many targets. Manipulate data and replicate to selected targets.	Using user-developed stored procedures.	Any required data transformation is performed by the user application.
Utilities supported for automatic loads of target tables	EXPORT/IMPORT, EXPORT/LOAD, LOAD FROM CURSOR option of the LOAD utility, depending on the target platform.	EXPORT/IMPORT, EXPORT/LOAD, LOAD FROM CURSOR option of the LOAD utility, depending on the target platform.	No automatic loads because there are no target tables.
Supported operating systems	Linux, UNIX, Windows, z/OS, iSeries	Linux, UNIX, Windows, z/OS	Linux, UNIX, Windows, z/OS
Supported relational database management systems	DB2 UDB, Oracle, Sybase, Teradata, Microsoft SQL Server	DB2 UDB	DB2 UDB
Administrative interfaces	Replication Center or command line	Replication Center or command line	Replication Center or command line

Event Publishing is primarily used as a capability to close the loop on a process, and can be used with other tools to automate processes for publishing DB2 information on an MQ queue. This enables a better response time, and small replication scenarios.

SQL Replication and Q Replication are the most frequently used replication approaches. We provide a comparison of the two approaches, depicted in Table 5-2.

For the comparison, we point the similarities and differences. As examples, we compare the:

- Infrastructure
- Sources and targets
- Data capture and apply

In addition, we provide an indication of which replication solution should be used in particular business intelligence scenarios.

Table 5-2 Comparison of SQL replication and Q replication

Points of comparison	SQL Replication	Q Replication
Location of the control tables for the capturing program	You must create the control tables for a Capture program on the DB2 UDB server where the Capture program runs. In most cases, this server is the same DB2 UDB server where the sources associated with the program are located.	You must create the control tables for a Q Capture program on the DB2 UDB server where the Q Capture program runs. This server is the same DB2 UDB server where the source tables associated with that program are located.
Number of capturing programs possible per DB2 UDB server	Multiple, each with its own set of control tables and identified by the schema of those control tables.	Multiple, each with its own set of control tables and identified by the schema of those control tables.
Location of the control tables for applying program	You can create the control tables for the Apply program on any server that can connect to the source server and the target server. In general, the control tables should be located on the server with the Apply program.	You must create the control tables for a Q Apply program on the DB2 UDB server where the target tables associated with that program are located.

Points of comparison	SQL Replication	Q Replication
Number of applying programs possible per DB2 UDB server	Multiple, each sharing one set of control tables and identified by a string called an Apply qualifier.	Multiple, each with its own set of control tables and identified by the schema of those control tables.
Source and target platforms	Sources and targets can be on the following relational database management systems: ► DB2 UDB for Linux, UNIX, and Windows ► DB2 UDB for z/OS ► DB2 UDB for iSeries ► Informix ► Microsoft SQL Server ► Oracle ► Sybase ► Teradata (for targets only)	Sources and targets can be on the following relational database management systems: ► DB2 UDB for Linux, UNIX, and Windows ► DB2 UDB for z/OS
Database objects that can be targets	DB2 UDB tables and views, and tables on non-DB2 relational databases.	DB2 UDB tables and stored procedures.
Pairing of sources and targets	You register a source. This information is stored in the Capture control tables. Then, you create one or more subscription-set members to map this registered source to targets. Information about these subscription-set members is stored in Apply control tables.	You create a Q subscription to map a source to a single target. No registration of the source is required. A source can be replicated to multiple targets by creating one Q subscription for each target. Information about the Q subscription is stored in Q Capture control tables and Q Apply control tables.
Grouping of source-target pairs	You group source–target pairs into subscription sets. Each source–target pair is referred to as a subscription-set member.	You can group Q subscriptions by replication queue map. No subscription set object exists in Q replication.

Points of comparison	**SQL Replication**	**Q Replication**
Subsetting of source columns and rows allowed?	Yes	Yes
Transforming data	You can transform data by using stored procedures or SQL statements run by the Apply program. On the source server, the stored procedures or SQL statements can transform staged data. On the target server, they can transform data in targets. The stored procedure interface limits transformations. The Apply program passes no parameters to a stored procedure and runs it once for an entire subscription set rather than for each subscription–set member. You can also use triggers to transform data that the Capture program writes to staging tables.	You can transform data by using stored procedures as targets. The Q Apply program calls the stored procedures and passes the changed data to them as parameters. This method of using stored procedures allows considerable versatility for data transformation.
Location of the capturing program	The Capture program runs on the DB2 UDB server where its control tables are located. In most cases, the source tables are also on the same server.	The Q Capture program runs on the DB2 UDB server where its control tables are located. The source tables are also on the same server.
How DB2 changed data is captured	The Capture program reads the DB2 recovery log and stores committed transactional data in staging tables.	The Q Capture program reads the DB2 recovery log and converts committed transactional data to messages.

Points of comparison	SQL Replication	Q Replication
How DB2 UDB changed data is transported	The Apply program fetches data from staging tables and applies the data to targets with DB2 SQL.	The Q Capture program puts data as messages in queues. WebSphere MQ moves the messages to the target system. The Q Apply program gets these messages from queues and applies data to the targets using DB2 SQL.
Location of the applying program	Apply programs can run on any DB2 UDB server in your network, provided that they can connect to the servers that contain the source, target, and Apply control tables.	Q Apply programs run on target servers.
How data is applied to DB2 UDB targets	The Apply program can process a subscription set in table mode or transaction mode. In table mode, the Apply program processes the fetched changes for each table separately, and then issues a single commit after all data is applied. In transaction mode, the Apply program applies the fetched changes to all of the target tables, in the order that the changes occurred in their corresponding transactions on the source server. The Apply program commits these transactions at transaction boundaries. You specify how many transactions to apply before each commit.	The Q Apply program can apply transactions concurrently whenever they do not involve dependent changes to the target tables. The Q Apply program uses various methods of conflict handling and resolution to make bidirectional and multidirectional replication possible.

Table 5-3 describes the most common scenarios for using SQL replication and Q replication, and recommends which technology best applies to each scenario. The recommendations are general in nature, and might not apply for your particular needs. As usual, a good understanding of your particular environment is required to pick the best approach.

Table 5-3 Common replication solutions

Configuration	Description of configuration	Recommendation
Data Consolidation (High Latency)	You can replicate data from many sources to a central repository. Data consolidation configurations are most often used to build data warehouses or operational data stores.	You can use either SQL replication.
Data Distribution (High Latency)	You can replicate data from a source to one or more targets that reside anywhere in a distributed network. Applications use the local target tables, so that they do not overload the network or central server. Each target server might only be interested in or allowed to access a subset of the data. You can choose to have a subset of columns and rows replicated so that each site sees only the data that it needs to see.	If you have a large number of targets, SQL replication might be the better choice because it allows you to stage data once and replicate that data to many targets but it will have a high latency.
Data Consolidation (Low Latency)	You can replicate data from many sources to a central repository. Data consolidation configurations are most often used to build data warehouses or operational data stores.	You can use Q replication.
Data Distribution (Low Latency)	You can replicate data from a source to one or more targets that reside anywhere in a distributed network. Applications use the local target tables, so that they do not overload the network or central server. Each target server might only be interested in or allowed to access a subset of the data. You can choose to have a subset of columns and rows replicated so that each site sees only the data that it needs to see.	You can use Q replication.

5.2 Messaging

In this section we describe update approaches using the messaging capabilities of WebSphere MQ and the DB2 Listener.

5.2.1 WebSphere MQ and continuous flow

WebSphere MQ messaging products enable application integration by enabling business applications to exchange information across different platforms, by sending and receiving data as messages.

Messaging middleware, such as WebSphere MQ, provides protection from losing messages by guaranteeing message delivery. With asynchronous messaging, the producer, consumer, and means of communication, do not have to be available at the same time, because the messaging software can safely pass the message from one party to another when they are available. The messages will be kept safe, and eventually processed with guaranteed delivery. However, in some situations, eventually may not be, satisfactory. Messages are often time-critical and need to be processed sooner rather than later, so the systems that process them must be highly available. Overnight batch processing, or spooling data that is ready for a future event, is typically not time-critical.

WebSphere MQ V5.3 is available for Linux for Intel®, Linux for z-Series, IBM eServer iSeries, and Microsoft Windows XP. They take care of network interfaces, assure once and once only delivery of messages, deal with communications protocols, dynamically distribute workload across available resources, handle recovery after system problems, and help make programs portable. So programmers can use their skills to handle key business requirements, instead of dealing with underlying network complexities.

Version 5.3 includes a number powerful features. As examples:

- Security using Secure Sockets Layer (SSL), the Internet standard for secure communication.
- High performance, especially for JMS (Java Message Service) applications, making WebSphere MQ the JMS provider of choice.
- System scalability and reliability, which is particularly useful for clustering of systems that can share workload.
- Simple for programmers to use API exits to allow monitoring or for implementing local standards.
- All WebSphere MQ products are euro compliant, and AIX, HP-UX, Sun™ Solaris and Windows versions are Tivoli Ready.

- WebSphere MQ messaging forms the key business integration layer of the WebSphere Software Platform.

WebSphere MQ provides a consistent multi-platform, application-programming interface. A key factor is time-independent processing. This means messages are dealt with promptly, even if one or more recipients are temporarily unavailable.

There are some important issues to consider when designing systems to deliver time-critical messages. First, the systems must be continuously available. Second, they must scale to the power required to handle the throughput. And third, they must be configurable for ease of control.

High availability sounds self-evident, but to process time-critical messages systems must be available and working at the time the message is sent. In the early days of message queuing, messages could be delayed in an offline system - even though they would be held safely until those systems became available again. The sending application did not have to check whether a message was processed — eventually it would be taken from the queue by the receiving application when the links were restored.

But with time-critical messages, that is not acceptable. WebSphere MQ clusters can choose from a number of different servers, so even if one is unavailable, there will be another to fill the need. Depending on how you specify the servers for the cluster, you can create an ultra-high availability environment for messages that must be processed immediately.

Servers need to process messages quickly, so there will be less need to hold the messages in a queue waiting for processing resources to become available. The alternative is a trickle feed, where messages are sent slowly but continually. You can obviously build systems that are powerful enough to process messages on demand, with minimum queuing, by buying faster systems, or by buying more systems, with WebSphere MQ clusters.

Having fewer but faster systems involves less administration but these systems often need more careful tuning to avoid contention and bottlenecks. With more systems and less sharing of resources, you have less contention for the resources within each system. Usually a balance exists where enough reasonably powerful systems are used, keeping administrative tasks at a minimum, yet giving enough parallel activity to complete all tasks in the required amount of time.

Configuration of a number of systems in even modest-sized networks can produce a high number of interconnections. The number of associated definitions, where parameters in one system must match or complement parameters in another, is also large. Controlling even a small number of multiple systems can produce too many combinations to effectively track.

Adding required definitions, making appropriate alterations as requirements change, and removing definitions that are no longer needed, are error-prone tasks. Often systems become unavailable due to errors introduced with these changes as the size and complexity of the systems environment increases. Using software such as WebSphere MQ clusters eliminates the need to make associated changes across the network of queue managers, helping reduce potential for errors.

Figure 5-4 shows a graphic depicting how WebSphere MQ is positioned to provide the integration required in a business environment. It enables different business applications to easily exchange information even when they all exist on multiple different platforms.

WebSphere MQ provides a common, Message Queue Interface (MQI) that enables application integration and makes programs portable. It enables business applications to exchange information — sending and receiving data as messages — across different platforms. A key feature of WebSphere MQ is time-independent processing. This means that messages are stored reliably for later delivery, even if one of the recipients is temporarily unavailable. Using WebSphere MQ to send and receive messages allows you to integrate applications on diverse platforms, enabling participation in business process integration.

The IBM WebSphere business integration portfolio delivers middleware that allows you to unite applications, people and business processes so they can work as one. Built on open standards, such as J2EE,XML and new Web services standards, this robust portfolio is endowed with reliability, scalability, and security.

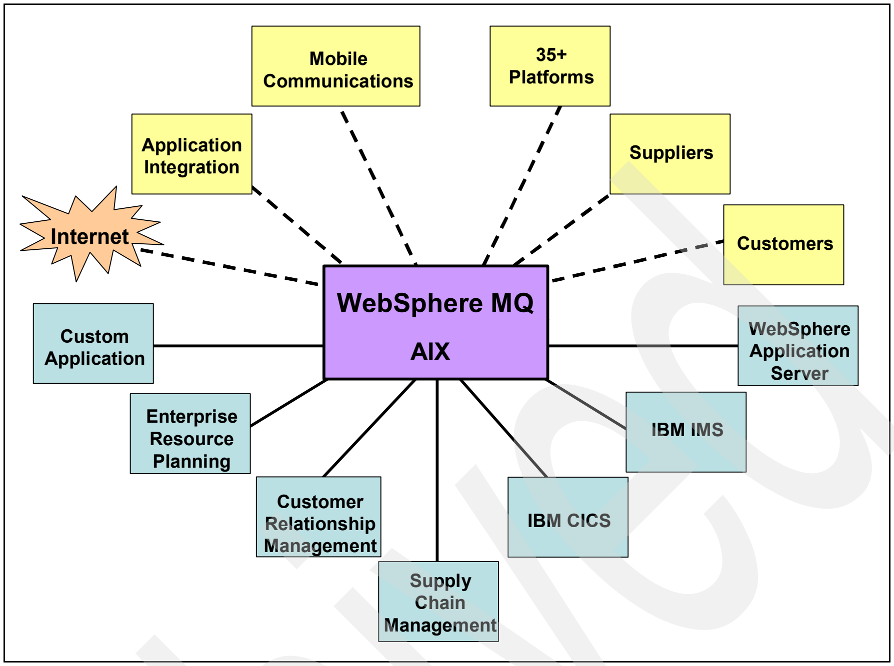


Figure 5-4 WebSphere MQ for business integration

5.2.2 DB2 MQ Listener

DB2 provides a capability to monitor a WebSphere MQ queue for messages. When a message arrives, a DB2 Stored Procedure can be invoked to process the message. This is called the DB2 MQ Listener.

You can configure the MQ Listener to listen to the WebSphere MQ message queues that you specify in a configuration database. MQ Listener then reads the messages as the arrive on the queue and calls a DB2 Stored Procedure with the message as an input parameter to be processed. The message are retrieved by priority order. If the message requires a reply, MQListener creates a reply from the output that is generated by the Stored Procedure.

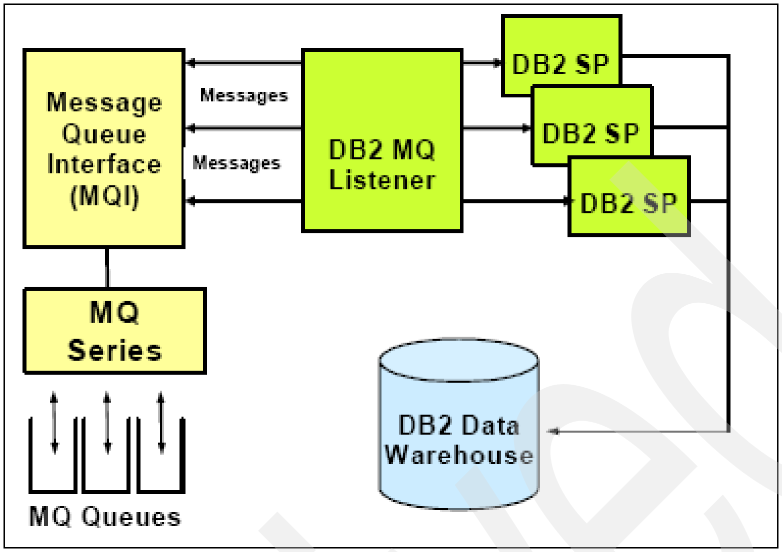


Figure 5-5 DB2 MQ Listener

MQ Listener runs as a single multi-threaded process. Each thread or task establishes a connection to its configured message queue for input. Each task also connects to a DB2 database on which to run the stored procedure. Each task will repeatedly execute the following until it gets a shutdown request.

- Get a message from MQ queue
- Call a user-supplied DB2 Stored Procedure, passing the message
- Sends reply (optional)
- Commit/Rollback the transaction which may be configured to include MQ and DB2 operations in the commit scope

In Figure 5-6 is an example of how the DB2 MQListener can be used to take messages off of a WebSphere MQ Queue and update the data warehouse. In this scenario, there may be multiple stored procedures updating one table. With multiple updaters on one table, there might have some undesirable locking situations. You also have to be careful in the number of stored procedures that the DB2 MQListener can start so as not to overwhelm the concurrently executing queries.

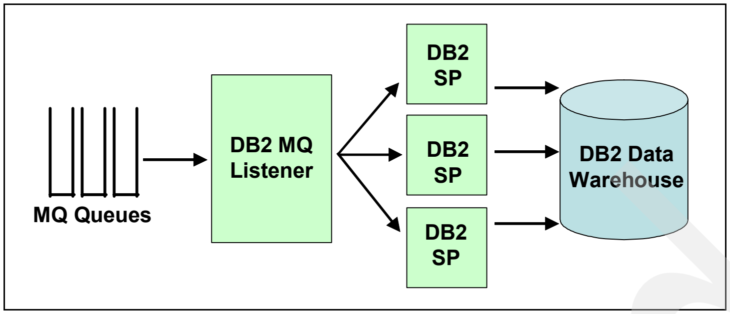


Figure 5-6 Using MQ Listener to transform and apply

If you implement the Transform function and the Apply function, as independent operations, you could use the DB2 MQListener to invoke a stored procedure that processes each message and uses the DB2 MQ functions, to put the processed record onto a WebSphere MQ Queue which is then fed to a continuously running stored procedure that applies those records to the data warehouse table, as depicted in Figure 5-7.

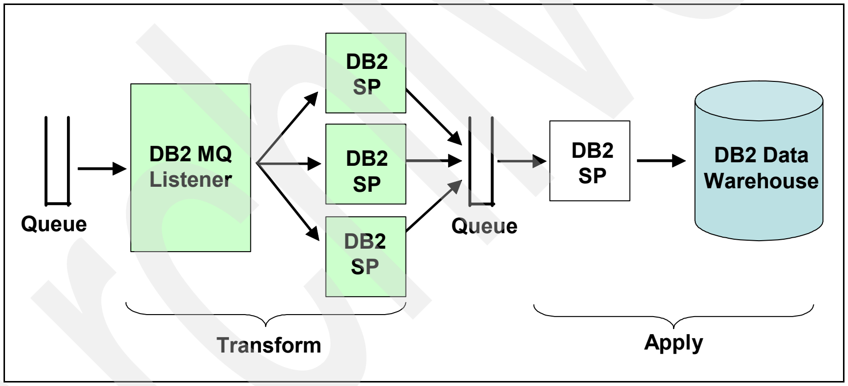


Figure 5-7 Separating the transform and apply function

This scenario provides several advantages. The time that it takes to perform the Transform function on a record is probably much longer that the time to actually apply the data to the data warehouse table. By separating the two functions, MQListener can invoke more parallel invocations of the stored procedure for the Transform function without affecting the concurrency of the Apply function. The WebSphere MQ Queue between the Transform and Apply functions provides a buffering mechanism to help smooth out the speed of update to the data warehouse.

5.2.3 DB2 MQ functions

DB2 provides a set of functions that allow us to read from and write to WebSphere MQ Queues using SQL without having to learn the WebSphere MQ Series application programming interface. These are implemented as a set of user defined functions (UDFs). This means that the support is available to applications that are written in any supported language that supports SQL. We can use these functions in SQL stored procedures and even from a DB2 command interface.

The DB2 MQ functions allow us to support various messaging paradigms, including Datagrams, Request/Reply and Publish/Subscribe. In the Datagram paradigm, we just send or receive messages with no expectation of a reply being generated. With a Request/Reply, when a message is sent, a reply is expected and a correlation ID is used to associate the reply with the request when it arrives back. With Publish/Subscribe, we just send a message to a topic and interested parties subscribed to that topic will receive the message. There can be any number of receivers for a topic.

In a basic configuration, a WebSphere MQ server is located on the database server machine along with DB2. The WebSphere MQ functions are enabled to DB2 and provide access to the WebSphere MQ server. DB2 clients can be located on any machine accessible to the DB2 server. Multiple clients can concurrently access the WebSphere MQ functions through the database. Through the provided functions, DB2 clients can perform messaging operations within SQL statements. These messaging operations allow DB2 applications to communicate among themselves or with other WebSphere MQ applications.

The **`enable_MQFunctions`** command is used to enable a DB2 applications to use the WebSphere MQ functions. It will automatically establish a simple default configuration that client applications can use with no further administrative action. The default configuration allows application programmers a quick way to get started and a simpler interface for development. Additional functionality can be configured incrementally as needed.

The Application Messaging Interface (AMI) of WebSphere MQ provides a clean separation between messaging actions and the definitions that dictate how those actions should be carried out. These definitions are kept in an external repository file and managed using the AMI Administration tool. This makes AMI applications simple to develop and maintain. The WebSphere MQ functions provided for DB2 UDB are based on the AMI WebSphere MQ interface. AMI supports the use of an external configuration file, called the AMI Repository, to store configuration information. The default configuration includes a WebSphere MQ AMI Repository configured for use with DB2 UDB.

Two key concepts in WebSphere MQ AMI, service points and policies, are carried forward into the DB2 MQ functions. A service point is a logical end-point from which a message can be sent or received. In the AMI repository, each service point is defined with a WebSphere MQ queue name and queue manager. Policies define the quality of service options that should be used for a given messaging operation. Key qualities of service include message priority and persistence. Default service points and policy definitions are provided and can be used by developers to further simplify their applications.

Queues can be serviced by one or more applications at the server upon which the queues and applications reside. In many configurations multiple queues are defined to support different applications and purposes. For this reason, it is often important to define different service points when making WebSphere MQ requests. This is demonstrated in the following example of a DB2 MQ function to send a message: **`VALUES DB2MQ.MQSEND ('ODS_Input', 'simple message').`** In this command, ODS_Input is the service point, and simple message is the message string being sent.

By using the WebSphere MQ DB2 functions, messages can be either received or read. The difference between reading and receiving is that reading returns the message at the head of a queue without removing it from the queue, while receiving operations cause the message to be removed from the queue. When you use a receive operation to retrieve a message, the same message can only be retrieved once. When you use a read operation to retrieve a message, the same message can be retrieved many times. The following example command demonstrate the retrieve (read) operation:

```
VALUES DB2MQ.MQREAD()
```

This sample operation returns a VARCHAR string containing the message at the head of queue that is defined by the default service using the default quality of service policy. If no messages are available to be read, a null value is returned. This operation does not change the queue. The following is an example of the receive operation:

```
VALUES DB2MQ.MQRECEIVE('Employee_Changes')
```

This operation example shows how a message can be removed from the head of the queue defined by the Employee_Changes service using the default policy.

Another feature of DB2 is the ability to generate a table from a User Defined (or DB2-provided) Function. You can exploit this table-function feature so that the contents of a queue are materialized as a DB2 UDB table. The following example demonstrates the simplest form of this feature:

```
SELECT t.* FROM table ( DB2MQ.MQREADALL()) t
```

This query operation example returns a table consisting of all of the messages in the queue that are defined by the default service and the metadata about these messages.

To return only the messages, the above example operation could be rewritten as:

```
SELECT t.MSG FROM table (DB2MQ.MQREADALL()) t
```

This support for WebSphere MQ Queues provides significant capabilities that can be utilized in developing a near real-time data warehouse.

In Figure 5-8 there are two examples of using DB2 MQ functions in stored procedures to implement the Transform and Apply functions of the near real-time data warehouse architecture.

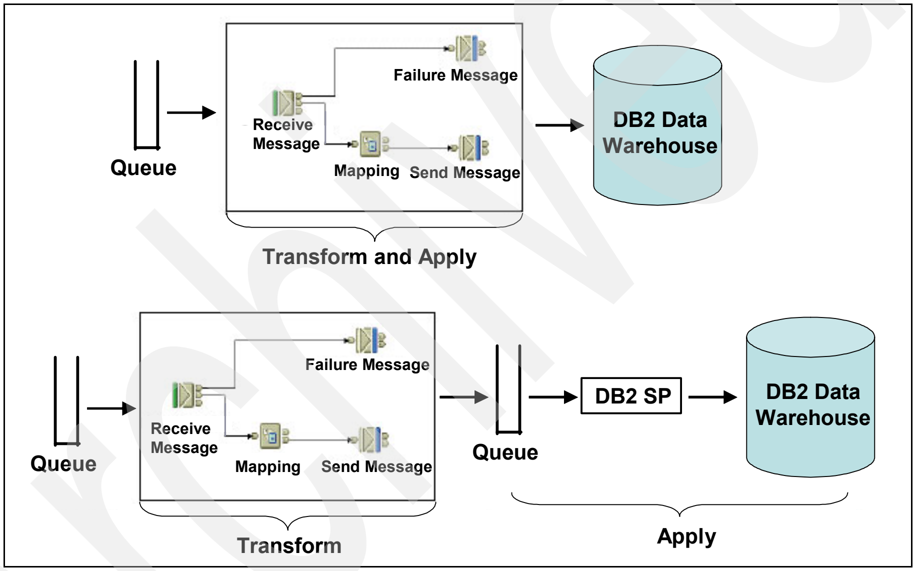


Figure 5-8 Using MQ functions in right-time data warehouses

These stored procedures can be simply developed using the SQL Procedure Language. Of course, you are not bound to just using DB2 MQ functions alone. This capability can be combined with other technology in developing a real-time data warehousing environment. Figure 5-7 on page 241 depicts the implementation of MQ Listener for the transform function along with the use of DB2 MQ functions in a stored procedure to implement the apply function. Of course, there are many other possible ways to develop these capabilities.

5.3 SQL generators and applications

Right-time data warehousing typically implies some type of asynchronous feeding of the data warehouse from the source systems that will involve some type of latency greater than zero - with zero being the optimum of real-time. As you move toward real-time data warehousing, you have to consider how to:

- Capture the data.
- Deliver that data to the data warehouse environment.
- Apply transformations to the data.
- Apply that data to the data warehouse with concurrent access.

You also have to consider the need to further feed the data to downstream aggregates or data marts.

In this section we introduce a new acronym, CDTA. It is generated from the ELT process (extract, load, and transform), and is used to describe the functional capabilities that will be needed to implement real-time data warehousing. They are Capture, Deliver, Transform, and Apply. These functions are depicted in Figure 5-9, as part of the data warehousing process flow. Ideally, these functions would be designed to be independent of each other. However, in an actual implementation, some of the functions may be combined, depending on the technology being implemented.

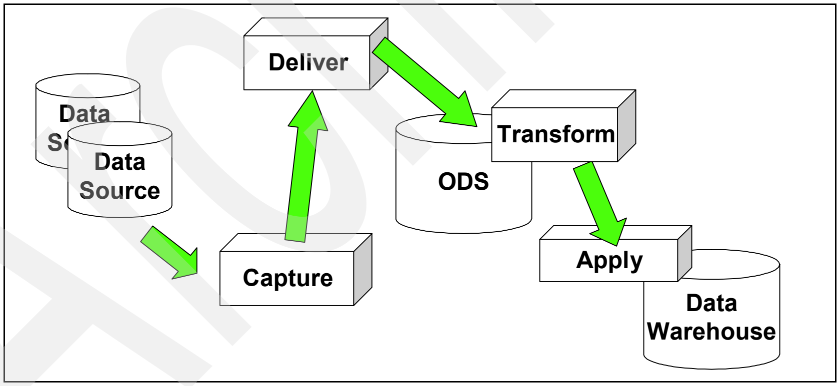


Figure 5-9 CDTA (ELT) data flow

5.3.1 Capture

There are many techniques that could be employed to capture data as it is created, some of which are more intrusive to the operational systems than others. The technologies employed in the capture function affects, or is affected by, the type of delivery capability that is available. Therefore, the capture function and the delivery function should go hand-in-hand.

One thing to investigate is whether to use the operational systems log or archive the changes that occur. Then the information can be captured from those logs or archives. For example, if the application maintains an audit trail, there could be a transaction that would be executed and pick up the data changes from the audit trail and provide them to the delivery function. If not, then perhaps changes could be made to the operational application system itself. For example, the application may provide the data changes to a delivery function, such as a message queue. This is depicted as capability 3 (the number inside the circle) in Figure 5-10.

If a relational DBMS is used for the operational systems, you can take advantage of capabilities such as triggers, or the DBMS change capture mechanism, to provide the delivery function with the data changes that occur. Database triggers work by taking an action whenever specifically defined changes to the data occur. However, triggers are executed, or fired, during the unit of work for the transaction that originally changed the data. This could possibly have an impact on the response time of these transactions. DB2 supports changed data capture by reading the database logs after the transaction has completed. See capability 1 (the number inside the circle) depicted in Figure 5-10. This will have less of an impact on the operational systems than the trigger technique. Typically these changed records will be stored in a relational table, but there is a capability to place these changes in a message queue. See capability 2 (the number inside the circle) depicted in Figure 5-10. There are also products from IBM that can capture changes in other non-relational database systems, such as IMS.

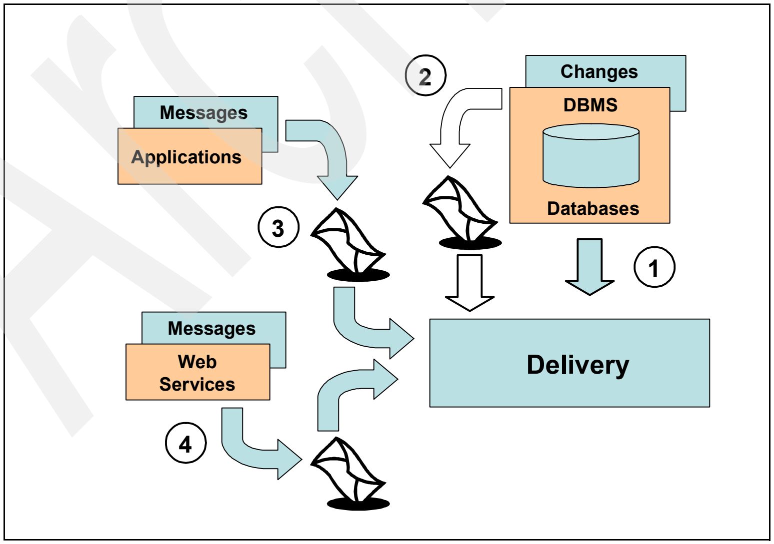


Figure 5-10 Capture function example

Although using Web services in the communication layer, this can vary upon your needs.There can be many variations in the way this capability is implemented. For example, there may actually be some form of this capability in your enterprise application integration (EAI) systems already. You may be able then to use those EAI systems to capture the data and route it to the data warehouse. What is important is that you are able to capture the required data as it changes and immediately provide it to some delivery mechanism. Another approach would be the use of Q Replication to deliver a low latency environment, as depicted in Figure 5-2 on page 227.

5.3.2 Delivery

So, how can you deliver the data to the data warehousing environment? One way would be to use some type of asynchronous delivery mechanism between the operational systems and the data warehouse environment. Two common techniques to accomplish this asynchronous communication are the use of relational tables and message queues. It is likely that both techniques may be deployed to meet various business and technological requirements. There is also a third mechanism which can be considered, but it only moves the data on-demand. That mechanism is federated technology. It can extend the access to data to include data that cannot, or should not, be kept in the data warehouse.

Relational tables

The fact that the use of relational tables is so pervasive and easy can help make the evolution to real-time a little easier. It is technology that we understand and that is supported by all the data management tools. However, the way in which it is used perhaps will be different.

The deliver function becomes a very simple matter of providing relational access to these tables. The tables could be local to the data warehouse environment or, more likely, remote. They might even be in different databases or different database management systems. If so, federation technology can be used to mask the fact that these might even be relational database products from different vendors. This will simplify the access by the transform function by allowing it to access one dialect of the SQL language and functionality.

Message queues

If you do not want to use relational, or if you do not want to physically write the data to disk, you can utilize asynchronous messaging middleware. You may already have this technology in your company in the form of Enterprise Application Integration (EAI) software, which is usually implemented with some type of messaging hub. You may even be able to capture data that is already flowing through the message hub to feed the data warehouse.

Whatever the technology, software, or application that is interacting with the message queue, it only sees a local queue onto which it places the message. Once the message is on the queue, the originating application does not know, and really does not care, about the eventual destination of the message. The messaging middleware will take care of delivering the message to the appropriate destination, in this case the data warehouse. The messaging middleware can directly forward the message to a queue in the data warehouse for point-to-point asynchronous communication, or it may actually be sent to a messaging hub first. This messaging hub is then responsible for routing the messages, perhaps to destination queues at multiple locations, and perhaps even reformatting the messages along the way.

Message queue middleware has been around quite some time, and is a very mature technology. It is available on almost every platform, and can take feeds from a variety of technologies - such as trading partners, Web services, enterprise applications, Web portals, mobile devices, and even telemetry sensors. For many years it has been a key technology in the approach of trickle feeding (continuous feeding) the data warehouse. Now that requirements are demanding more than batch processing, messaging is becoming a primary choice for continuously feeding updates to the data warehouse. It is a very powerful approach. And, with business performance monitoring coming to the forefront of requirements, businesses are beginning to use messaging technology to feed the data warehouse in order to provide the kind of near real-time information that they need for decision-making.

With all the capabilities and low latency of Q Replication, the adoption for this technology as the delivery layer is recommended.

On-demand

The deliver function can deliver data to the data warehouse, but it can also deliver data directly to users. This is because some data may never be placed in the data warehouse, even though it is still required by users. All the data can be made available for query as one means of delivery. Another form of proactive delivery is the analytic application, as depicted in Figure 5-11. These are applications that analyze the data, and perform actions based on a set of predefined rules.

There is a world of data that could be useful in your analytical applications, but which, for many reasons, cannot be physically stored in the data warehouse. Perhaps the requirement is only for occasional access and you cannot justify developing a real-time data feed for it. Maybe it is unstructured data, such as text files, e-mails, images, and videos that would take too much storage space. Perhaps we need information from large external data sources such as from various search engines which would be absolutely impractical to store locally.

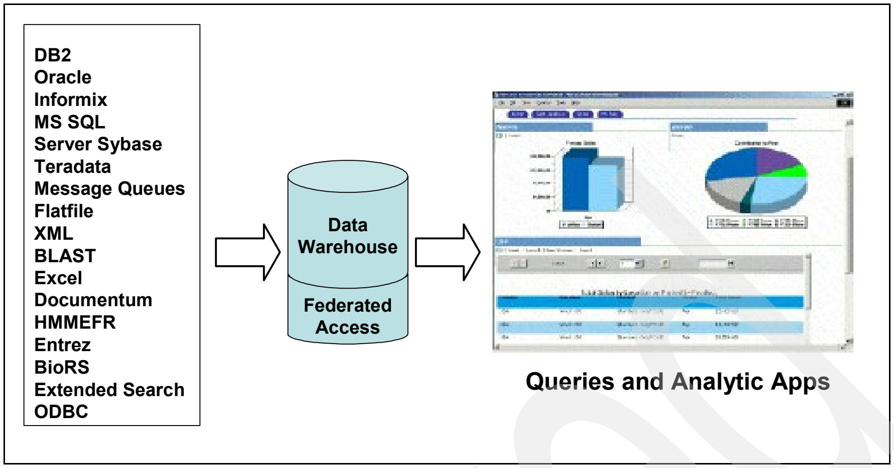


Figure 5-11 On-demand data access using federated technology

That is where data integration technology comes into play. This technology provides data access to much more than relational, or even local, data. Once federated access is set up for these various data sources, they appear as relational structures that you can access as though they were part of the data warehouse, and by using the same SQL language used to access the tables in the data warehouse.

5.3.3 Transform

Transformations cover the gamut from simple data type conversions, to heavy duty cleansing, to very complex auditing requirements. You have to very carefully weigh the transformation requirements against the latency requirements. The smaller the latency required, the fewer transformations you will be able to accomplish. If you absolutely have to perform a significant amount of data transformation, then you may have to adjust the latency requirement. For example, transformations may have to be batched to enable appropriate calculations. Or, you may need to calculate total metrics for the batch to enable derivation of percent-of-total for each item.

The transform function is a fairly independent approach from the first two functions of capture and deliver. You can capture and deliver the data to the data warehouse environment in a near real-time fashion, but you can also collect or stage this real-time data and process it in batches. Or you can actually do the opposite. You may still capture and deliver the data in batches, and then process the batches as soon as they arrive. The key is to get the data into he data warehouse quickly. Of course the ultimate approach is to capture and deliver the data in near real-time, and transform and also apply the data in real-time. Or, you may utilize several of these approaches in combinations as you incorporate real-time capability into the data warehouse population systems.

One of the first things that you need to do is to functionally separate the Transform function from the actual update of the data warehouse tables, which is the primary purpose of the apply function. The transform function should feed the apply function but both should operate independently, as depicted in Figure 5-12. This can be done by placing some type of data buffer between them.

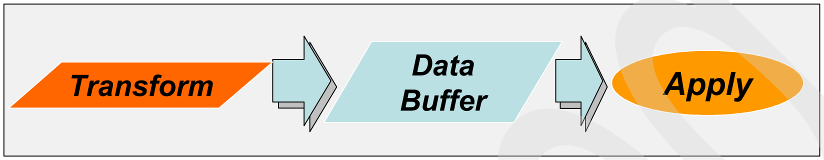


Figure 5-12 Transform and Apply functions

The transform function would create changed data records that could very simply be applied directly to the data warehouse tables. These records would already have been transformed, cleansed, surrogate keys applied, and so forth. The record would also contain an indication of whether it is to be an update to an existing data row or inserted as a new data row. In the case of a dimension table, you would have a series of updates and inserts, depending on the type of dimension update strategy you employ. Of course, for fact tables, you would only have inserts.

One of the primary caveats of real-time data warehousing is that you will need to update the data warehouse while users are concurrently executing queries and performing analyses against it. By keeping the apply function independent from the transform function, you can have more control of what techniques are actually used to accomplish this balance between update and query.

There are two basic alternative architectures for the transform function, batch and continuous. A batch architecture is very similar to what is included with most of ETL systems today. That is, the process lets the data accumulate and then it is processed as a collection of data updates.

There are many factors that may lead us to more of a batch type transform function. First of all, the data may be delivered in batches. Some of the required transformations or auditing functions can only be performed on a batch of data. The data could be supplied in typical batch fashion or may be provided continuously - but then be staged in a message queue or relational table until ready to be processed as a batch. These batch transform functions may be scheduled to run at some periodic interval ranging from every minute or so to every few hours, while some will be automatically invoked, on-demand, based on the arrival of specific data or conditions.

There are businesses today using a batch method in which they receive batch files around the clock, and must get that data into the 24x7 data warehouse within a certain time after receiving the data file. This is most definitely a form of near real-time data warehousing.

The continuous flow architecture is similar to what is done today in an EAI system. When a data record, probably in the form of a message, is available, it is acted on as soon as it is received. There is typically some type of application that is looking at, or listening to, a message queue. Then, when a message appears on the queue, it will be acted upon. There are many different approaches, from an implementation viewpoint, for detecting that a message has appeared in the queue and how it should be processed. While some mainstream ETL processes can access message queues as a source, some of these tools may still have to operate on a batch basis. However, this is the area where EAI tools, or message brokers, excel. They can take a message from a message queue, process it, and put it into a queue and/or a relational table. However, these EAI tools are sometimes limited in the transformations they can perform. The trend is to blend the capabilities of these two types of tools.

5.3.4 Apply

Relational database technology has matured to the point that it is now very possible to implement continuous update scenarios using a variety of technologies and techniques depending what tools are available and what kind of storage mechanisms are used for the data buffer between the transform function and the apply function.

If you use memory techniques, such as pipes, you can have an application that is constantly executing. It can take changed data from the pipe and apply it directly to the target table. This same technique can be used with Q replication, or even a database table, as the data buffer. These programs can even be implemented as stored procedures, and, if the DBMS supports it, can use SQL to access messages in a message queue.

There are also capabilities being added to database management systems that can be used. As an example, consider bulk load utilities. Bulk load utilities at one time were not very intelligent. Their task was to take data from a flat file and put it into a relational table by building the data pages directly.

However, the bulk loaders now have much more capability and intelligence. For example, they can load data, in parallel, to the various partitions of a partitioned database. They can take input from a named pipe, or from a database cursor that allows data to be selected directly from a table or view, and feed it directly to the load utility (without first being written to a file or table). Also interesting is that this view could actually be a view created over a message queue.

5.3.5 ELT in an enterprise business intelligence system

ELT (extract, load, and transform) can be used in several scenarios, because it is tightly integrated with the enterprise needs. In Figure 5-13 you can see a scenario that mixes other technologies but can take advantage of the ELT processes.

Some of the benefits of ELT are:

- Does not rely on proprietary language
- Reduced costs
- High level performance
- Learning curve is short due to the usage of already known SQL functions.

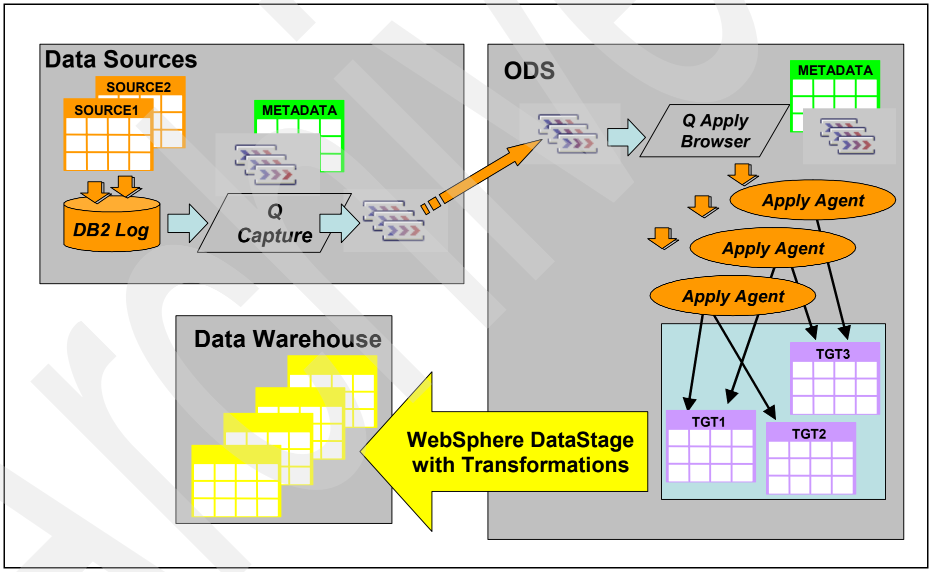


Figure 5-13 ELT on the Enterprise Business Intelligence system

5.4 Extract, transform, and load (ETL)

In this section we describe a data warehouse update approach using ETL, which is provided by WebSphere DataStage. The ETL process consists basically of three major steps: data extraction, transformation, and loading. The primary purpose for an ETL tool in a continuous flow scenario is to integrate the data access, transformation and cleaning, as needed.

The ETL tool should be able to connect to every information source needed to build your data warehouse. By using data flows you will be able to map and transform the data to satisfy the requirements of your enterprise business intelligence system.

The data warehouse updates may occur on the movement from the ODS to the data warehouse, or by using another staging area, depending on how sensitive the systems must be regarding concurrency. An ETL process is depicted in Figure 5-14.

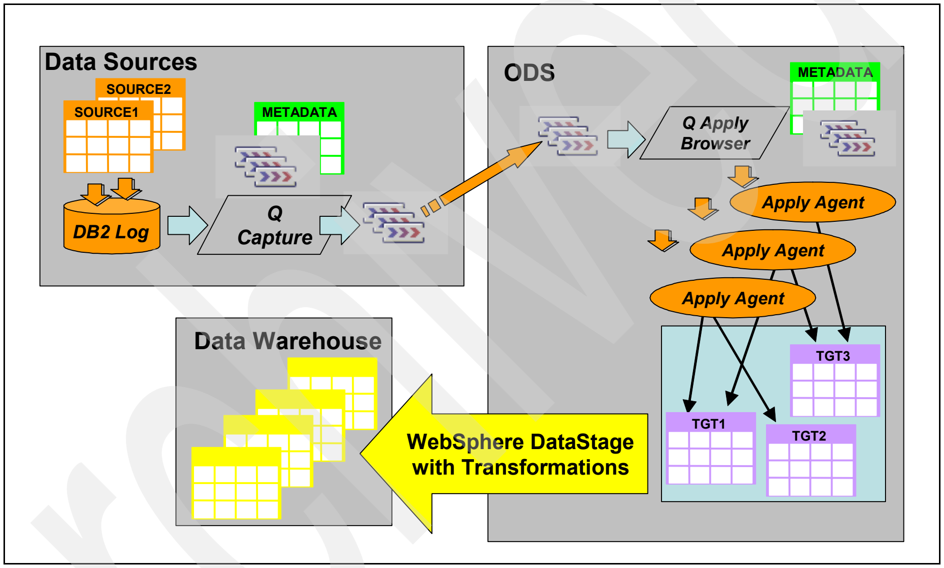


Figure 5-14 ODS and data warehouse in an ETL process

5.4.1 Using ETL tools

The use of an ETL tool can help to address some issues such as:

- ► Project control and management
- ► Maintenance
- ► Access control and security
- ► Rapid development

Project control and management

ETL can help on project control and management because the flows can be documented as they are developed. Typically ETL tools create macro flows in a graphical way, making it easier to understand and resulting in an accurate data flow diagram.

Maintenance

Because the flows can be developed with a graphical tool, they are much easier to maintain. And, they can be used to analyze and check exceptions, and even monitor data flows.

Access control and security

ETL tools typically can provide a central view of users and rights. And, it is typically the responsibility of the tool to control authentication and authorizations.

Rapid development

Developing data flows using a graphical mapping tool significantly improves the data integration flow creation.

WebSphere DataStage plays a very important role in the IBM strategy for the real-time enterprise. It is responsible to extract, transform, and load the data according to the real needs of the business intelligence environment, relying on on a code-free and GUI-based development to deliver a real-time solution.

The flows in DataStage are created as jobs, with data flowing between them, as depicted on Figure 5-15. The job is basically transforming data from a text file to another text file. It is a very simple job but shows how the data is moved from one point to another. Jobs should be created to move and transform data based on your particular needs.

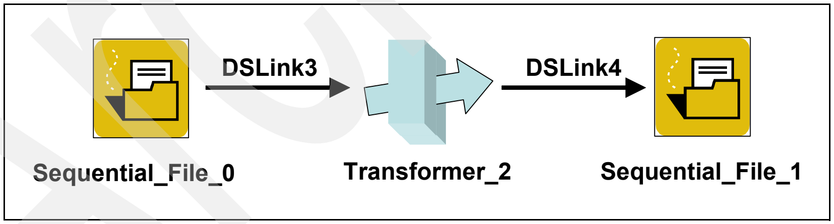


Figure 5-15 A simple DataStage job

By using an ETL tool in the data flow, and having a batch insertion into the data warehouse, does not preclude you from implementing a continuous update process. The ODS should be populated on a near real-time schedule, providing up-to-date analytical data to serve both operational and tactical purposes. In addition, this set of data can be used as an additional data source for strategic decision-making.

5.4.2 A WebSphere DataStage project

A WebSphere DataStage project is comprised on the following components:

- DataStage jobs
- Built-in components
- User-defined components

Jobs

A DataStage job consists of a series of individual stages, linked together to describe the flow of data from a data source to the data warehouse or other data target. Each stage describes a particular phase of the process. For example, one stage may extract data from a data source, while another transforms it. Stages are added to a job and linked together using the same DataStage Designer.

You must specify the data you want at each stage, and how it is handled. For example, do you want all columns in the source data or only a particular subset? Should the data be aggregated or converted before being passed on the next stage?

Data properties are defined by:

- Table definitions: These specify the data you want, and each table definition contains:
  - Information about the table or file that holds the data records.
  - A description of the individual columns.
- Data elements: Each data elements describes one type of data that can be stored in a column. The data element associated with a column defines the operations that can be performed on that column. DataStage has numerous predefined data elements representing commonly required data types, such as date, time, number and string. You can also define your own special data elements.
- Transforms: These convert and cleanse the data by transforming it into the format desired and defined for your data warehouse. DataStage provides a large library of built-in transforms to get you started fast in this phase.

Together, these properties determine what occurs at each stage of a DataStage job. The properties are set up project-wide and are shared by all the jobs in a project.

DataStage supports three types of jobs:

1. Server jobs are both developed and compiled using DataStage client tools. Compilation of a server job creates an executable that is scheduled and run from the DataStage Director.

2. Parallel jobs are developed and compiled using DataStage client tools. Compilation of parallel job creates an executable that is scheduled and run from the DataStage Director. However, parallel jobs require a UNIX server for compiling and running. Parallel jobs support parallel processing on SMP, MPP, and cluster systems.
3. Mainframe jobs are developed using the same DataStage client tools for server jobs, but compilation and execution occurs on a mainframe computer. The Designer generates a COBOL source file and supporting JCL script, then enables uploading them to the target mainframe computer. The job is compiled and run on the mainframe computer under the control of native mainframe software.

When a job runs, the processing stages described in the job design are performed using the data properties defined. Executable jobs can be packaged for use on other DataStage systems.

Stages

In DataStage there are two types of stages:

1. Built-in stages: Supplied with DataStage and used for extracting, aggregating, transforming, or writing data. These stages can be either passive or active.
2. Plug-in stages: Additional stages defined in the DataStage Manager to perform tasks that the built-in stages do not support.

A stage usually has at least one data input and one data output. However, some stages can accept more than one data input, and output to more than one stage.

Stages can be passive or active:

- Passive stages define read and write access to data sources and repositories: Types of passive stages are:
  - Sequential
  - UniVerse®
  - ODBC
  - Hashed
- Active stages define how data is transformed and filtered. Types of active stages are:
  - Transformation
  - Aggregation

Stages and links can be grouped to form a container. A container is represented by a container stage:

- Grouping stages together into containers can make the design simple, and easy to read.

- Links to and from a container are represented by container input and container output stages.
- Containers can be shared and re-used between other jobs, so you do not have to recreate the same process twice.
- Shared containers can be thought of as common library of objects.

Server components

DataStage has three server components:

1. Repository: This is a central store that contains all the information required to build a data mart or data warehouse.
2. DataStage Server: The server runs executable server jobs, under the control of the DataStage Director, that extract, transform, and load data into a data warehouse.
3. DataStage Package Installer: This is a user interface for installing packaged DataStage jobs and plug-ins.

Client components

DataStage has four client components:

1. DataStage Manager: A user interface used to view and edit the contents of the repository.
2. DataStage Designer: A graphical tool used to create DataStage server and mainframe jobs.
3. DataStage Director: A user interface used to validate, schedule, run, and monitor DataStage server jobs.
4. DataStage Administrator: A user interface used to set up DataStage users, control purging criteria, and install National Language Support (NLS) maps and locales.

The DataStage Manager

The manager is a graphical tool that enables easy viewing and managing of the contents of the DataStage Repository. It is used to browse, import, and edit meta data about the data sources, targets, and transformations. It also allows you to define or install DataStage.

The DataStage Designer

The Designer is used to build jobs by creating a visual design that models the flow and transformation of data from the data source through to the target data warehouse. The Designer graphical interface lets you select stage icons, drop them onto the Designer work area, and add links. They you can define the required actions and processes for each stage and link, and compile the server jobs and check the debugger as required. This is depicted in Figure 5-16.

A job created with the Designer is easily scalable. This means that you can easily create a simple job, get it working, then insert further processing, additional data sources, and so on.

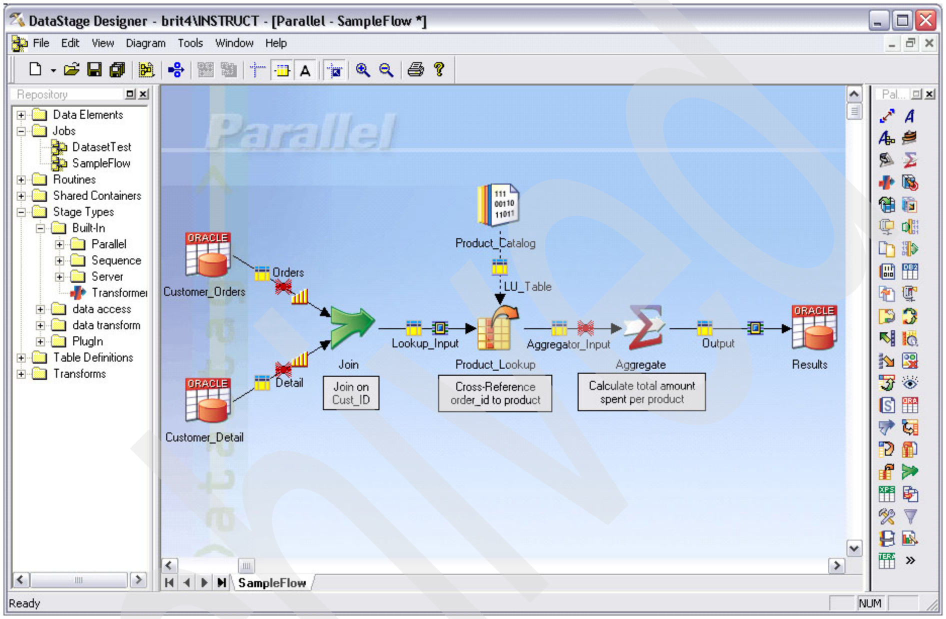


Figure 5-16 DataStage Designer

The DataStage Director

The Director enables you to run, monitor, and control server jobs built in the Designer. When you run a job, the Director prompts for any runtime parameters. It enables monitoring of the run-time performance, and events such as error conditions. You can run server jobs on an ad-hoc basis or schedule them.

The DataStage Administrator

The Administrator enables you to set up DataStage users, control the purging of the Repository, and, if installed, manage locales.

Types of data sources and targets

Data for server jobs belongs to one of the following categories:

- ODBC Data: This is data is defined as any data source that can be accessed through the Open Database Connectivity API. This is accomplished by means of database-specific ODBC driver. The list of databases for which ODBC drivers are available is quite large, giving DataStage the ability to handle data from a wide variety of databases.

- Sequential File Data: Sequential file data is also known as text files or flat files. These can be of fixed-length or delimited format. As long as you tell DataStage what you are using for a delimiter, it can be whatever you choose.
- Direct Access Data: Direct access data is accessed directly without using SQL. DSEngine, UniVerse, hashed file and UniData® databases, are examples of direct access data.

6

The project test environment

In this chapter, we discuss and describe how to create a real-time enterprise. However, it soon becomes a very particular job because it depends on the particular enterprise requirements. So, in Chapter 7, “The case study” on page 281, we provide an example implementation of a real-time enterprise in the retail industry. The example scenario gives you a closer look at how to implement such an environment using IBM approaches and solutions.

We describe the software and hardware used in our particular case study, and provide a macro view of how the IBM products in the environment relate. Then we describe two test scenarios that have a lot, and also very little, in common. We elected to have two scenarios because of the timing of this project. A key WebSphere product was in development that we wanted to use, but the project schedule dictated our particular selections.

The product was WebSphere Message Broker. The current version was V5, and the one in development was V6. The V6 product contained very important new functionality called the enterprise service bus (ESB). To emphasize this new functionality we decided to run two test scenarios for the case study, making use of both WebSphere Message Broker Version 5 and WebSphere Message Broker Version 6.

6.1 Preparing for the case study

The test scenarios described in this chapter were built to develop and execute the redbook case study. As mentioned, we used two versions of WebSphere Message Broker. It is for that reason we needed two test scenarios. We named the two test scenarios, simply test scenario A and test scenario B.

In test scenario A, we used WebSphere Message Broker Version 5 to integrate the information in the enterprise. In the second test scenario, which we call test scenario B, we used WebSphere Message Broker Version 6 (with the Advanced Enterprise Service Bus). The ESB aggregates the ESB functions and integrates with several other applications that are not Web services enabled.

6.1.1 Business functional areas

The following are brief descriptions of the business functional areas included in the test scenario, along with descriptions of the particular hardware and software used to implement them.

CRM System

A CRM system has been implemented to manage the company customer base. It holds all the data relative to each customer, including purchases, preferences, and interactions. This data is used to establish a good relationship with customers, enhance their purchasing experience, and create a data base for mailings and specific promotions.

The system is built on DB2 UDB, and runs in the Linux operating environment. The system was developed using the following hardware and software configurations:

- Hardware:
  - Intel server with 1 processor
  - 1024 MB memory
  - 20 GB disk space
- Operating System:
  - Red Hat Enterprise Linux 3
- Software:
  - DB2 UDB V8.2 Enterprise Server Edition FP9A
  - WebSphere Information Integrator - Advanced Edition
  - WebSphere MQ Express V5.3 fixpak 6

Inventory system

The inventory system maintains the inventory of all retail products offered by the case study sales organization.

This system also runs in a Linux operating environment, and is uses an Oracle database system. It was developed using the following hardware and software configurations:

- Hardware:
  - Intel Server with 1 processor
  - 1024 MB memory
  - 20 GB disk space
- Operating System:
  - Red Hat Enterprise Linux 3
- Software:
  - Oracle 10g

Supplier management system

The supplier management system maintains information about all the suppliers used by the organization. This includes information such as, location, contact information, products supplied, pricing, delivery methods, times, and cost, business volumes, and evaluation.

The system runs on a WebSphere Application Server and interacts with the other systems in the test scenario using Web services interfaces. It was developed using the following hardware and software configurations:

- Hardware
  - Intel Server with 1 processor
  - 1024 MB memory
  - 1 x 20 GB disk space
- Operating System
  - Windows 2000 SP4
- Software
  - DB2 UDB V8.2 Enterprise Server Edition FP10
  - WebSphere Application Server V5

Purchasing System

The purchasing system enables and tracks all purchases by the organization. It gathers and evaluates available offerings to help determine the best price and terms available for potential purchases. Once determined, purchase orders are created, sent to the supplier, and the goods are shipped. Purchase order status is tracked and monitored until the order is received and payment made. A history is kept on all purchases to enable purchase trending, quality levels, satisfaction, and pricing.

The system runs in a Microsoft Windows 2000 operating environment, and is built on an SQL server database. It was developed with the following hardware and software configurations:

- Hardware
  - Intel Server with 1 processor
  - 1024 MB memory
  - 20 GB disk space
- Operating System
  - Windows 2000 SP 4
- Software
  - Microsoft SQL Server 2000

Sales system

The sales system tracks the sales of all products by the organization. The sales price, quantities, and discounts are traced for history, comparisons, and calculation of profit levels. The purchaser is captured and tracked, and the system interfaces with the CRM system. This provides the sales department with information needed for such activities as resales, promotions, discounting and advertising.

The system runs in a zSeries operating environment, and is built on a DB2 database. It was developed with the following hardware and software configurations:

- Hardware
  - zServer with 2 processors
  - 2048 MB memory
  - 15 GB disk space
- Operating System
  - z/OS R7
- Software
  - DB2/390

6.1.2 The big picture

The test scenario is comprised of business functional areas, the test environment flow, and the interaction between them. This is depicted in Figure 6-1. It represents the overall architecture to integrate all the business functional areas, provide real-time updates to the data warehouse using WebSphere MQ for data transfer, generate alerts based on key performance indicator thresholds, display alerts to a management dashboard with DB2 Alphablox and the WebSphere Portal Server, and initiate action to resolve the alert. The result is a system that not only provides business intelligence, but also represents a closed-loop, on demand real-time enterprise.

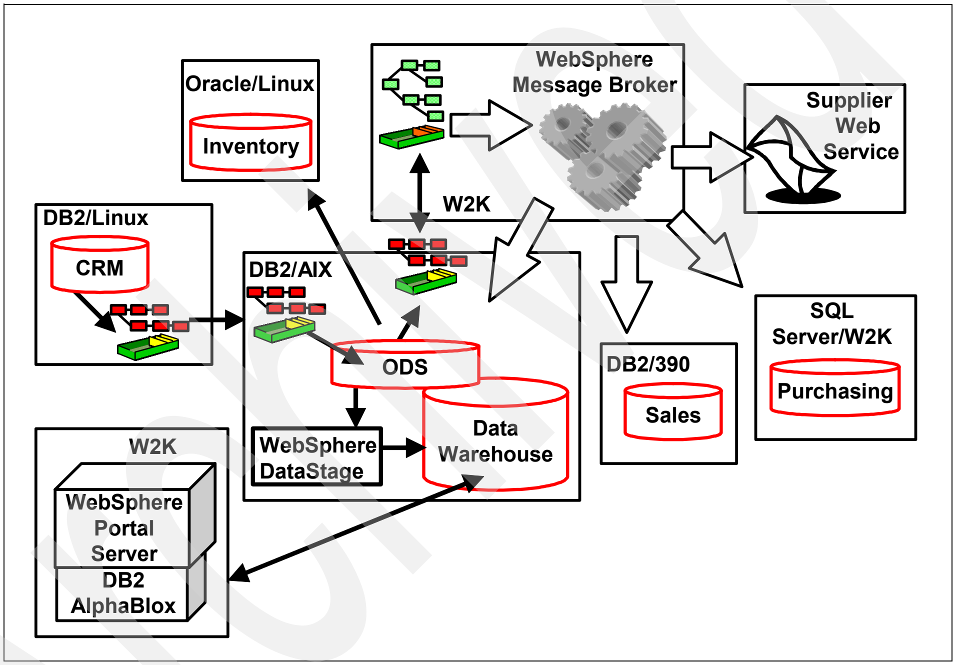


Figure 6-1 Big Picture

The results of testing, detailed message and data flow, considerations and conclusions are summarized in Chapter 7, “The case study” on page 281. Here we provide a description of the retail scenario and the test environment in which it was developed.

6.2 Test Scenario A

This test scenario is designed around a fictitious retail company called ValueTrend. ValueTrend is a leading retailer in its market, selling a variety of products through multiple channels. These channels consist of a chain of traditional supermarkets and large hyper-market stores - the ValueTrend Superstore. ValueTrend also has self-service Web-based eBusiness solutions, and their mission is to deliver high quality products and deliver high customer value and services.

The test scenario is described in Chapter 7, “The case study” on page 281. Briefly, it describes how the company is able to detect and resolve a product issue in a proactive manner. This is enabled by their implementation of a real-time enterprise. Now we can take a look at the environment used in the case study.

6.2.1 Software overview

In Figure 6-2, we depict a high-level overview of the test environment. It shows the applications used, and the data and process flow through the organization. A messaging system is used to communicate between the business functional areas. Data is stored in the data warehouse, and used to support queries, and the business dashboard. These elements support the needs of the company for business intelligence.

The architecture was structured and developed to satisfy the basic requirements of a real-time enterprise environment. In that environment, we have included several business functional areas that can be found in most retail companies. The business areas, and associated systems, included are:

1. Customer Relationship Management (CRM)
2. Inventory Control
3. Supplier Systems
4. Purchasing
5. Sales

To satisfy the requirements of the operational environment, we defined and developed the following systems management capabilities:

1. Data Warehouse
2. Front-end portal and dashboard
3. Message management

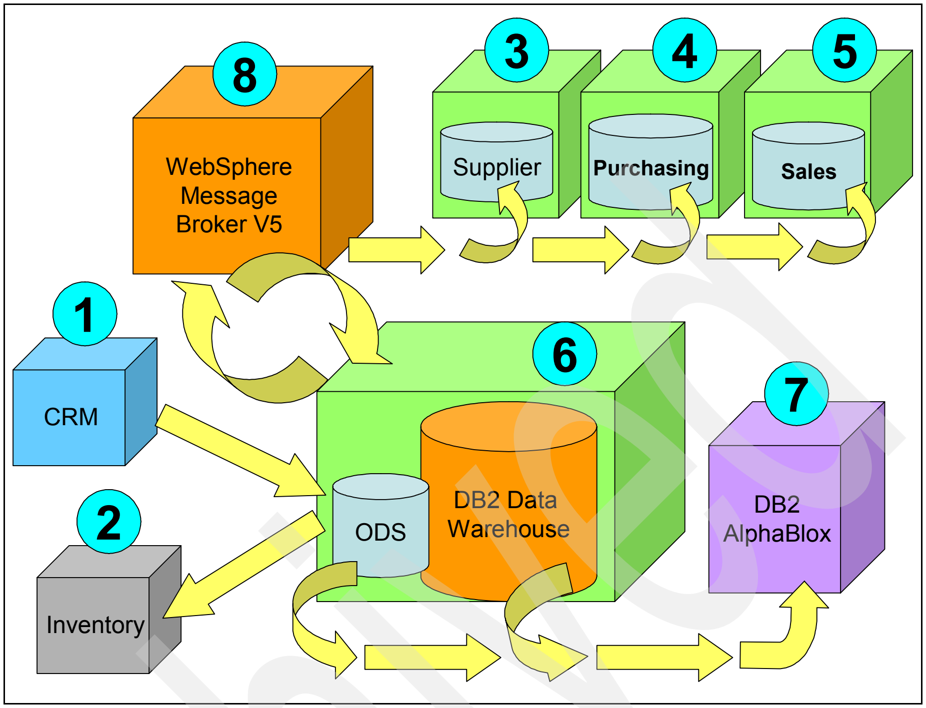


Figure 6-2 Test environment

To implement functionality and management systems, we used the following products:

- DB2 UDB V8.2 Enterprise Server Edition
- WebSphere MQ V5.3
- WebSphere Information Integrator V8.2 Advanced Edition
- DB2 Connect V8.2 Personal Edition
- WebSphere Message Broker V5
- WebSphere DataStage V8
- DB2 Alphablox V8.3

The environment was created for functional testing only. In an real business solution, you would consider additional capabilities and variables, such as the data volume, complexity of transactions, and performance requirements, to architect your particular solution environment.

In the next section we provide additional details about the systems, applications, and products used in the test scenario.

6.2.2 Test environment

We have described the business functional areas that are included in the test environments. In this section we describe the data flow and the products that enable it. This scenario data flow is depicted in Figure 6-3. The data warehouse server is hosting the operational data store (ODS) and DB2 data warehouse. This is the enterprise data warehouse environment, which holds the enterprise data used for monitoring, tracking, analyzing, reporting, and decision making. DB2 Alphablox is the front-end capability for those activities.

Alphablox is responsible for development and visualization of dashboards that enable the management team to monitor activities of the organization. This includes such capabilities as enabling alerts, event processing, graphs, and reports. This is the product that provides managements view into the enterprise.

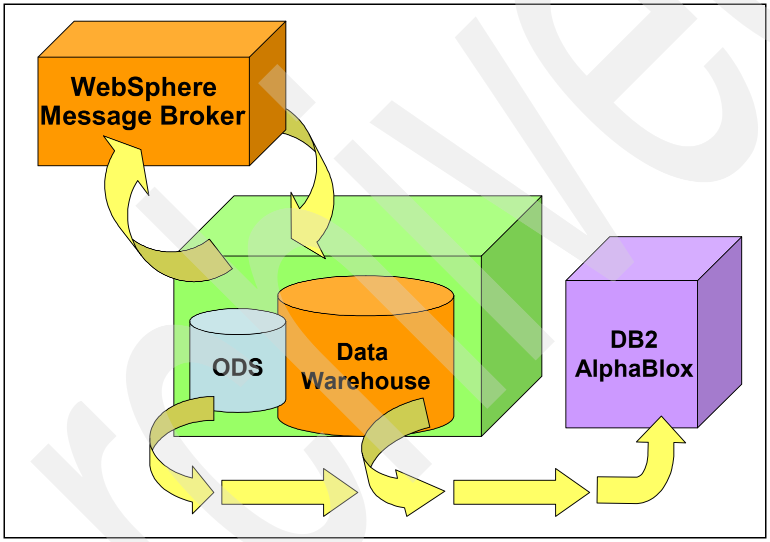


Figure 6-3 Scenario data flow

Alphablox can also enable the next step in the flow, which is to start corrective action when operational problems are detected. For example, it can automatically begin the execution of an analytic application, notify management team when key performance indicators dictate delivery, and enable analysis to find the answers they need by supplying the appropriate data. Performing these actions is, in effect, enabling a closed-loop environment for the enterprise.

WebSphere Message Broker is the product responsible for the systems integration and communications. Messages are sent to update the data warehouse and to enable inter-application communications. Data is sent in the form of a message, and the Message Broker guarantees message delivery. It is the glue that holds the systems together.

WebSphere Message Broker V5

WebSphere Message Broker V5 has a special role in the test scenario. It is responsible for enabling the interaction between all the business functional areas, and for updating the data warehouse with information about those interactions. It interacts with WebSphere MQ, which is the messaging product responsible for passing messages throughout the system.

WebSphere message broker has a number of key capabilities, such as:

- Distributing information and data generated by business events in real-time to people, applications, and devices throughout the enterprise - and beyond.
- Managing information flows between independent systems housed in separate divisions and departments.
- Extending the reach of a business beyond the firewall by supporting multiple transport protocols.
- Increasing business flexibility and reducing development costs by separating integration logic from applications.
- Improving the flow of information around the business, moving away from hard-coded point-to-point links to more flexible distribution mechanisms such as publish/subscribe and multi-cast.
- Facilitating communication among otherwise incompatible applications.
- Eliminating tedious and error prone integration programming logic by performing any-to-any data transformation in-flight within the connectivity layer.
- Integrating multiple applications, networks and device types using an advanced enterprise service bus that enables the conducting of business reliably and securely.

The WebSphere Message Broker uses the following hardware and software configurations:

- Hardware
  - Intel Server with 1 processor
  - 1024 MB memory
  - 20 GB disk space
- Operating System
  - Windows 2000 SP 4

- Software
  - DB2 UDB V8.2 Enterprise Server Edition FP10
  - WebSphere MQ V5.3 FP 11
  - WebSphere Message Broker V5 FP 6

Data warehouse environment

The data warehouse environment is the base element for the BI infrastructure supporting all business functional systems. It stores and manages all the enterprise information used for decision making. It was developed with the following hardware and software configurations:

- Hardware
  - RISC with 8 processors
  - 32768 MB memory
  - 300 GB disk space
- Operating system
  - AIX 5.2
- Software
  - DB2 UDB V8.2 Enterprise Server Edition FP10
  - WebSphere Information Integrator Advanced Edition
  - WebSphere MQ V5.3 FP 11
  - WebSphere DataStage V8

DB2 Alphablox

DB2 Alphablox is the product responsible for retrieving data from the enterprise data warehouse and presenting it to the requestor. That requestor can be a person, an analytic application, or some Alphablox capability.

A person can make specific requests for data for data analysis, and it will be provided by Alphablox. This can also happen automatically. For example, based on detection of some event threshold, an alert can be posted (an Alphablox capability) or an analytic application can be automatically dispatched.

DB2 Data Warehouse Editions V8.2.1 includes DB2 Alphablox. It integrates the Web-based analytics and reporting of DB2 Alphablox with the data mining, online analytical processing (OLAP), and extract, transform, and load (ETL) already available. The addition of DB2 Alphablox to the Data Warehouse Edition (DWE) provides a comprehensive BI platform operating on common data structures within the data warehouse.

DB2 Alphablox has an open architecture that is built to integrate with existing IT infrastructures. That can help lower the total cost of ownership by letting you leveraging existing resources and skill sets. In addition, it is optimized for rapid application delivery and deployment, and provides a component-based, comprehensive framework for integrating analytics into existing business processes and systems. By implementing solutions that include DB2 Alphablox, you can increase the value of your information assets by delivering business insight to the right people at the point-of-decision.

The Alphablox environment was developed with the following hardware and software configurations:

- Hardware
  - Intel Server with 1 processor
  - 1024 MB memory
  - 20 GB disk space
- Operating System
  - Windows 2000 SP 4
- Software
  - DB2 UDB V8.2 Enterprise Server Edition FP10
  - DB2 Alphablox V8.2
  - WebSphere Application Server V5
- WebSphere Portal Server V5

6.3 Test Scenario B

This test scenario is basically the same as scenario A, except that we used some enhanced functionality in WebSphere. That came in the form of the WebSphere Message Broker V6 and the Advanced Enterprise Service Bus (ESB). This extends the integration of a service-oriented architecture to encompass the entire enterprise and beyond.

6.3.1 Software overview

In this scenario, we enabled the ESB and created a service oriented environment. This service oriented test scenario is depicted on Figure 6-4.

Basically this scenario is the same as Scenario A. The only difference is that now, the closed-loop structure is accomplished through the use of the ESB. So, we now get all the benefits of ESB, and have a solution that is significantly more flexible, and more cost effective to implement.

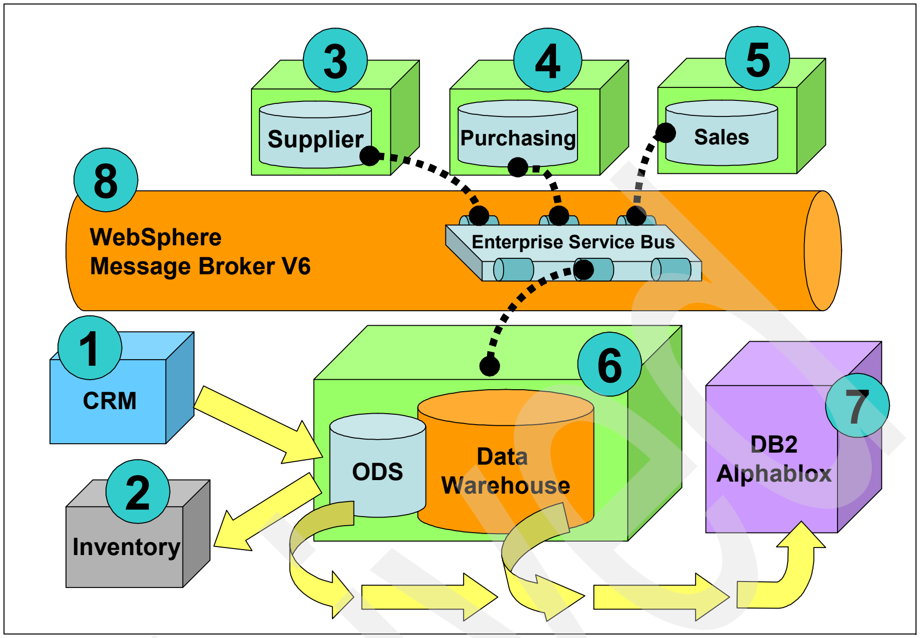


Figure 6-4 ESB/SOA enabled test scenario

6.3.2 Test environment

In this section we describe the data flow and the products that enable it. As mentioned in the previous sections, this scenario is basically the same as Scenario A, except we have added the ESB. The environment and data flow is depicted in Figure 6-5.

The enterprise service bus is the responsible for the systems integration and communications. Services are executed to integrate the data warehouse and to enable inter-application communications.

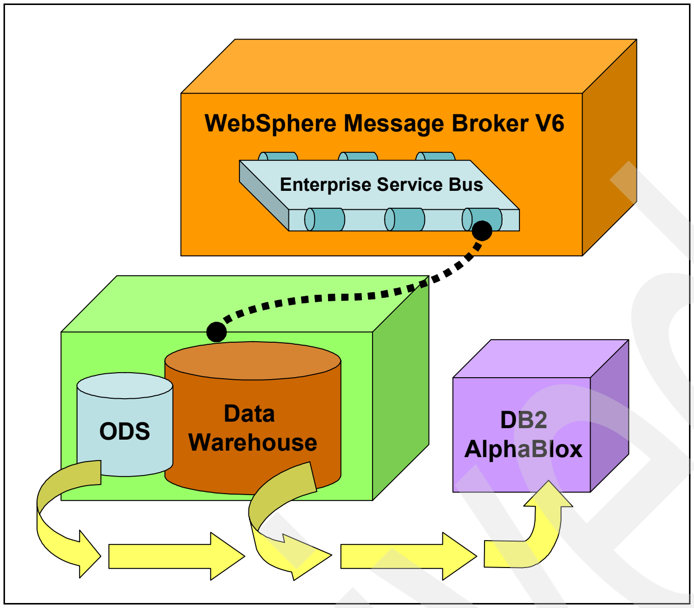


Figure 6-5 Test environment flow - ESB Scenario

WebSphere Message Broker V6

In a service oriented architecture, WebSphere Message Broker can be the connectivity layer. It optimizes information distribution between service requesters and service providers. As examples, it can:

- Distribute any type of information across and between multiple diverse systems and applications, providing delivery of the right information in the right format at the right time.
- Reduce the number of point-to-point interconnections and simplify application programming by removing integration logic from the applications themselves.
- Route information in real-time to any endpoint using a powerful publish-and-subscribe matching engine.
- Validate and transform messages in-flight between any combination of different message formats, including Web Services, other XML and non-XML formats.
- Route messages based on (evaluated) business rules to match information content and business processes
- Improve business agility by dynamically reconfiguring information distribution patterns without reprogramming end-point applications.

- Use access control to securely deliver personalized information to the right place at the right time.

The WebSphere ESB uses the following hardware and software configurations:

- Hardware
  - Intel Server with 1 processor
  - 1024 MB memory
  - 20 GB disk space
- Operating System
  - Windows 2000 SP 4
- Software
  - DB2 UDB V8.2 Enterprise Server Edition FP7
  - WebSphere MQ V6
  - WebSphere Message Broker V6

We have demonstrated two approaches that can be used in implementing a real-time enterprise. Both accomplish the same task, but Scenario B uses the latest technology with the ESB. The result is significant savings in development costs, a more flexible implementation, and a solution based on standards.

6.4 The real-time enterprise - a case study example

In this section we provide a high level overview of the case study described, implemented and demonstrated in Chapter 7, "The case study" on page 281. There we illustrate a real-time enterprise example scenario regarding a retail organization that sells appliances.

The environment

In the example scenario, customers can purchase appliances from this retailer from a Web site, by telephone, and directly from one of their retail outlets. The suppliers for the retailer publish their current inventory information via a Web service which the retailer has integrated into their business application environment, called the Retailer Appliance System. Figure 6-6 depicts this system with the customer and supplier channels.

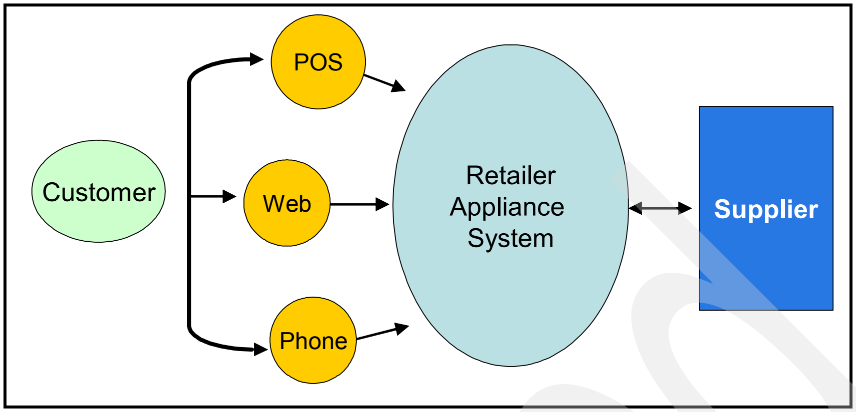


Figure 6-6 Appliance retailer with customer channels.

The retailer replenishes their inventory through Web services published by their suppliers. The suppliers have also provided Web service interfaces to their assurance database. The information flow of the business processes for the retailer is illustrated in Figure 6-7.

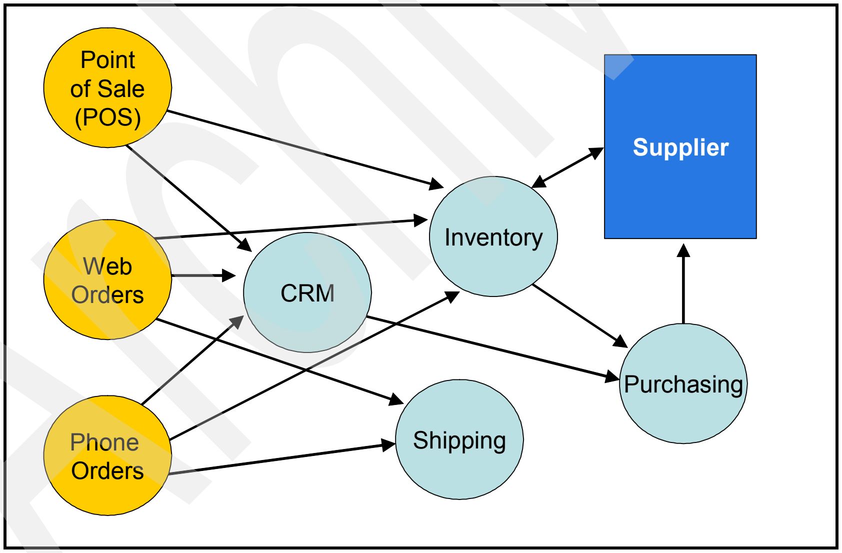


Figure 6-7 Retailer functional flow

When a customer transaction occurs from any of the three customer channels, information about the transaction is captured by the retailer's CRM and operational systems. The interaction of the operational, analytical and data warehousing systems is shown in Figure 6-8.

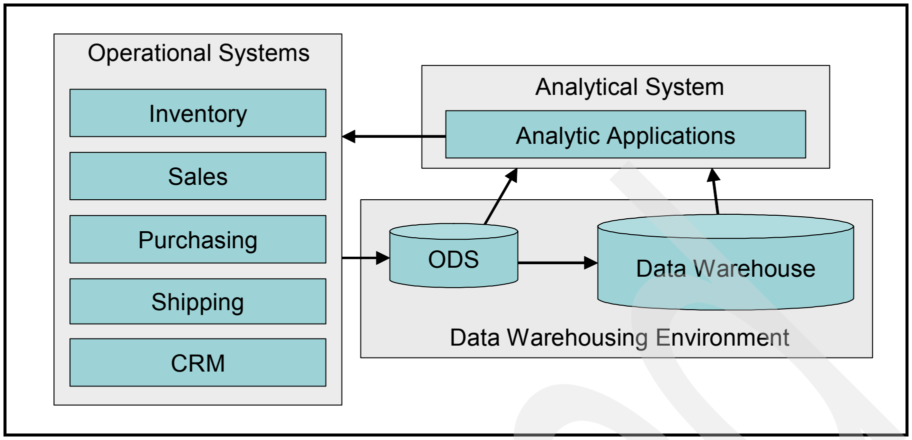


Figure 6-8 Enterprise environment

Information from the operational systems is populated to the operational data store (ODS). The ODS is populated frequently, and so contains current operational data, except for the CRM data which is transferred to the ODS once a week. The retailer has BI applications, in the analytical system, that analyze the sales, inventory and other data in the data warehousing environment (ODS and data warehouse) to monitor and manage the business. In addition the customer has applications that analyze the CRM data to monitor and manage the CRM operational environment.

When an item is purchased, information about both the item and purchase transaction is recorded in the operational systems. If the purchase is through one of the retail stores, the sales, inventory and CRM data is updated in the operational systems. If the purchase is made by phone or through the internet, shipping data is recorded in addition to sales, inventory and CRM data. The customer channels interact with the operational systems as depicted in Figure 6-9.

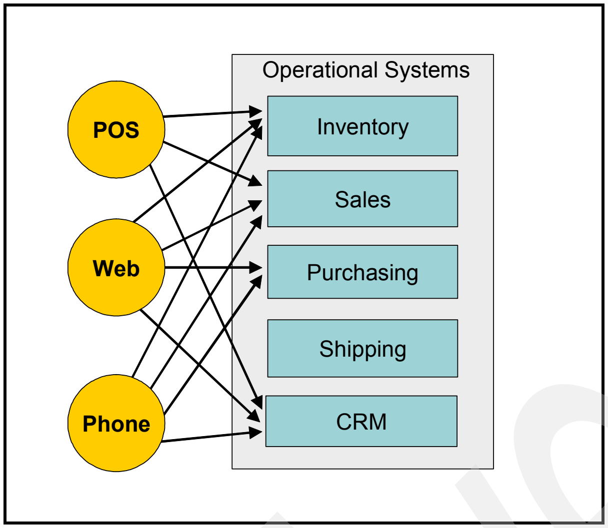


Figure 6-9 Operational systems and sales channels interactions

The CRM system for the retailer captures information about each customer that purchases an item. The system also captures information when an item is returned. When an item is returned to a particular store or via an internet or phone order, the retailer updates the operational systems with the transaction. In addition, the retailer captures information about why the item was returned in the CRM system.

When a customer returns an item they are asked to provide a reason for the return, and, in addition, the retailer asks the customer to fill out a voluntary questionnaire. If the customer returns an item via the internet, they are asked to fill out a form, and are provided with an item return authorization. Once the return is complete, the sales and inventory systems are updated. The CRM system is updated with the information collected through the channels.

The retailer proactively manages the business to enable meeting performance measurements and goals. To do this, performance targets (or key performance indicators - KPIs) have been set, such as total sales and total returns. And, thresholds have been determined to use with the monitoring system. When thresholds are exceeded, alerts are generated to enable management to take prompt corrective action. Several key performance indicators (KPI) have been selected for use, and they are gross sales per item, total gross sales, and total returned items. Other important KPIs include sales by channel, profit, profit by product, profit by channel, inventory, inventory carrying cost, and shipping costs.

6.4.1 The scenario

The following scenario description illustrates the real-time enterprise system at work. On a particular day, an alert appears on the management dashboard. It indicates the level of product returns, for a particular product, has exceeded the defined threshold. In addition to the visible alert, an analytic application is deployed. On determining the particular product that has caused the alert, the application notifies the sales and shipping systems, and places a 24 hour hold on all sales and pending shipments for the item. It also sends an email to the procurement manager, as illustrated in Figure 6-10.

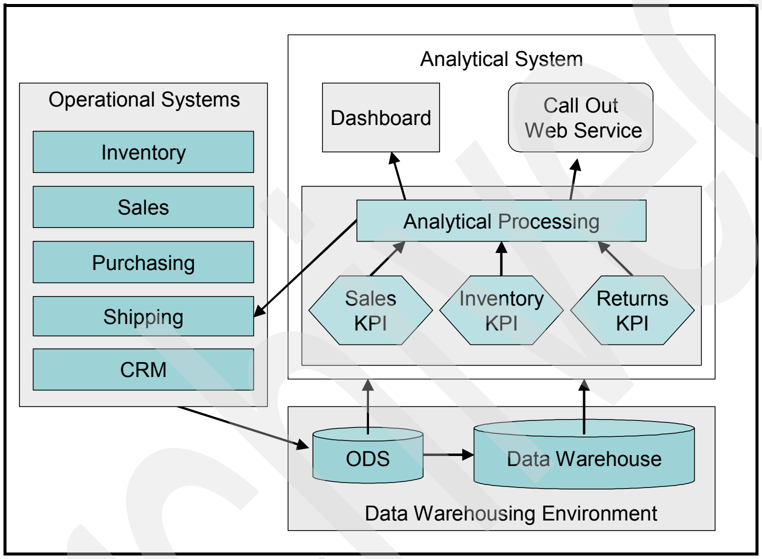


Figure 6-10 Analytical system information flow

The procurement manager reviews the information provided by the analytic application, and determines that one particular brand of electric kettle has a high rate of return due. The reason codes in the return data indicate that there could be a potential defect in the kettle. The manager initiates a query to the quality assurance system of the supplier of that kettle, who notices that a problem has already been identified with that particular kettle. The manager contacts the supplier and discovers that several retailers have been returning this particular item, for the same reason.

The supplier initiates an action to investigate, and the results indicate that there may in fact be a dangerous defect. The decision is made to recall those kettles. The relationship between the retailer and the supplier systems is illustrated in Figure 6-11.

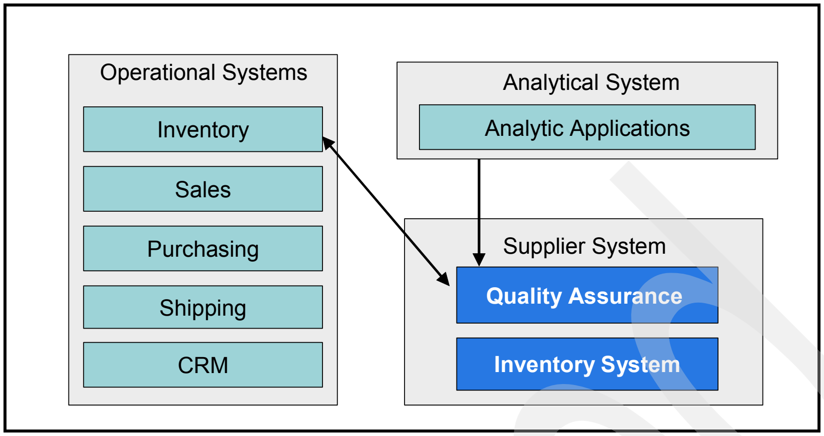


Figure 6-11 Retailer supplier system interaction

The procurement manager changes the hold placed on the kettles that have not yet been shipped, to a *stop shipping* status. Next, notices are sent to customers who ordered, or already purchased, a kettle. The procurement manager then begins to assess the impact of the recall by using the analytical systems. During this investigation it is noted that the CRM data is populated to the data warehouse only once a week. That means the data they are using could be as much as a week old, and so the problem could be more widespread than the data currently available indicates. The manager determines that they need to correlate the sales and CRM data, to get an accurate picture of the problem.

In addition to resolving this problem, management decides they must make additional changes to the system to minimize the recurrence of this type of problem. They want the system to be capable of avoiding such problems, rather than just minimizing the impact once it is discovered. Since they already have a fairly sophisticated system, the changes made are reasonably minimal. In this case it is decided that the CRM data will update the ODS twice a day, instead of once a week. Having data that is more current will enable potential problems to be recognized more quickly and enable action to be taken much sooner. With this new capability the retailer will be able to minimize loses, which impact profitability, and maintain a high level of customer satisfaction and loyalty.

The impact of the real-time enterprise

The above scenario represents a very simple example used to illustrate several of the characteristics and advantages of the real-time enterprise. One key characteristic of real-time enterprise is application integration. This is illustrated in the example by the supplier and the retailer sharing information. It is this sharing of information that enables the retailer and the supplier to quickly determine that the item is defective and should be recalled.

If the retailer did not have access to the supplier inventory and quality control systems, the procurement manager would have had to request the information about the item from the supplier, and consequently it would have taken much longer to understand the scope of the issue. So the real-time enterprise extends beyond the enterprise organizational boundaries.

Another characteristic of the real-time enterprise is the timely availability of data. Although the retailer had sufficient analytics to detect the issue, they did not have the latest CRM information available and consequently detected the issue a week after the KPIs for sales and inventory became critical. If they had detected this issue earlier, the retailer could have reduced the impact on their customers and sales.

The real-time enterprise is an environment where an analytic application can effect a change in the operational environment. This is illustrated by the hold on the orders that have not yet been shipped for the item. The hold reduces the business impact of the recall, by preventing the shipment of items that will later need to be recalled. This saves the retailer the cost of the initial shipments and the cost of the return of the recalled item.

The real-time enterprise has enabled the retail management to be proactive with regard to possible issues that effect their business. By designing an environment where the application responded to an event by engaging the procurement manager, the retailer was able to determine that they had a recall situation before the manufacturer had determined that the product needed to be recalled. The availability of the transformed current and historical data provided the procurement manager with easy to interpret information, and consequently enabled them to make correct and timely decisions.

This is a real-time enterprise in action.

7

The case study

In this chapter we describe a case study for a real-time enterprise. We developed this case study with a number of IBM products and technologies. The technologies used are all described in more detail in other chapters of this redbook. In this chapter we put them into action.

The products and technologies used represent significant capabilities. We highlight and use some of those capabilities, but certainly not all of them. That would be well beyond the scope of this redbook.

So we focus only on a subset of the capabilities, but that itself is significant. In this case study, we demonstrate how a real-time enterprise enables us to proactively meet the goals and business measurements of our enterprise. It includes the use of real-time data, to perform right-time business intelligence, for recognition and resolution of issues that could prevent the attainment of our business goals. It is a real-time closed-loop environment that is the business advantage.

It is this proactive nature of problem prevention and avoidance that enables the optimization of business resources and attainment of the business measurements and goals.

7.1 Introduction

In this section we present a case study, which is based on a fictitious retail company. We use this as a way to demonstrate how a real-time enterprise might operate. In particular, we describe the ability to detect a problem and correct it before it can impact the business measurements. It is this proactive ability that enables avoidance of problems, rather than just the minimization of their impact. This type of environment that enables management to really manage the business.

7.1.1 The company

ValueTrend is a leading retailer in its market, selling a vast variety of products through a number of channels. These channels consist of a chain of traditional supermarkets and large hyper-market stores - the ValueTrend Superstore. ValueTrend also has a self-service Web-based eBusiness solution. The ValueTrend mission is to deliver high quality products, and high customer value and services.

ValueTrend looks at many ways to measure and analyze sales data daily and thereby strives to be proactive and flexible when planning their purchase and sales strategy. At the same time they look at being operationally flexible, enabling them to react fast to market changes and competitive situations. There goal is to respond within days, rather than weeks or even months.

Areas where Valuetrend desires to react quickly is in customer complaints and product quality issues. The goal is to maintain their image as the leading high quality retail company in their marketplace.

ValueTrend and the real-time enterprise

By implementing facets of a real-time enterprise, ValueTrend aims to achieve better overall customer value and satisfaction. These goals will be achieved through the real-time enterprise by better and faster reaction to changing customer needs, continued efforts to reduce their cost structure, and continually looking at improved services to maintain their competitive advantage.

The key customer satisfaction goal is focused on having the ability to react more quickly to market changes, and by delivering the best product quality.

ValueTrend products

ValueTrend deals with both food and non-food products. In this case study, we focus on one product only and use it to develop the case study scenario involving a product life-cycle and quality issue.

7.1.2 The data flow

Figure 7-1 depicts the initial data flow through a subset of some of the sub systems that exists in ValueTrend - our fictitious retail company. The numbering in the Figure 7-1 relates to the case study described below.

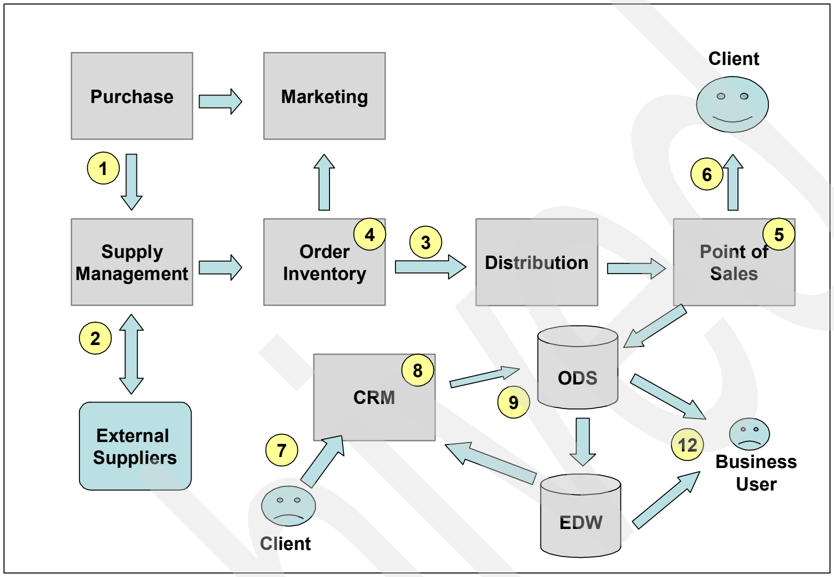


Figure 7-1 The ValueTrend original data flow

Case study

The following is a brief description of the events and activities involved in the case study:

1. A purchase order for a ValueTrend hair-dryer is placed by the central purchasing department.
2. The order for the hair-dryer is fulfilled by the supplier and delivered to the ValueTrend central distribution center.
3. The hair-dryer is then distributed to the ValueTrend retail store.
4. The Inventory system is updated to reflect this event.
5. The hair-dryer is now available to satisfy incoming sales orders from the ValueTrend sales channels.
6. Time passes and orders are processed. This should typically result in products being delivered and the result a happy customer.
7. ValueTrend receives calls from a number of unhappy customers complaining about the hair-dryer. They all indicate a belief that it is defective.

8. All returns, and a code indicating the reason for the return, are registered in the CRM system. Records of the returns are created in the CRM system, and are then used as measures for determining customer satisfaction. The reason codes have been implemented to handled all direct customer dialogs. All product return records are given a status code reflecting the seriousness of the defect.
9. The return reason information is collected and transferred to the ODS by a weekly batch ETL process.
10. The returns information is, at a scheduled point-in-time extracted, transformed, and loaded into the ValueTrend EDW.
11. The central returns department receives the actual hair-dryers back from the stores, and, with their BI capability, discover that a high percentage of the sold hair-dryers are being returned.
12. An investigation is launched by quality control. Sales statistics are produced by taking information from EDW and the ODS.
13. It is decided by ValueTrend to recall the hair-dryer. Through a press release, and stores notice boards, customers are informed that they can return the ValueTrend hair-dryer and get a full refund.

During the time span from when customers first began returning the hair-dryers, to the point-in-time when the decision is made to remove them from the store shelves and the eBusiness catalog, many days has passed. And in that time, a large number of hair-dryers had been sold.

Because of the long period of time taken by ValueTrend to respond to the problem, the company was exposed to a high risk of loosing credibility and thereby also potential loss of revenue on similar products.

It was determined that an improved process to handle these types of issues needed to be designed and deployed - quickly.

Improving the data flow

The objective of the improved process is to achieve the data flow through the processes with much lower latency. This is a much discussed issue in many companies today. How can access to real-time data about the processes be made available and result in improving the processes and minimizing their cost?

In this case study, we show a simple automated closed-loop process. It illustrates how a number of technologies, and approaches, can be used, along with process changes that enable better integration with those technologies, to enable a real-time enterprise environment. With the real-time data, companies can be more proactive. That is, trends and issues can be discovered and resolved more quickly, thus avoiding, or at least minimizing, their impact. This can result in a lower cost across the entire business value chain, improved customer satisfaction, and the resulting business advantage.

The overall objective of the case study is to highlight how we can achieve the closed-loop capabilities with a number of different products, techniques, and technologies available in the IBM integration portfolio.

All the operational business systems each play a vital role in the day-to-day operation of the retail business example. The Business Intelligence systems, for example, play both a strategic and operational business decision role as well, as well as being the key component in the closed-loop real-time enterprise case study.

Although the case study certainly does not address all aspects of the data flow in any real retail company, even this data can have a much wider of use than what is being illustrated. The details of the data flow in ValueTrend improved process, is depicted in Figure 7-2.

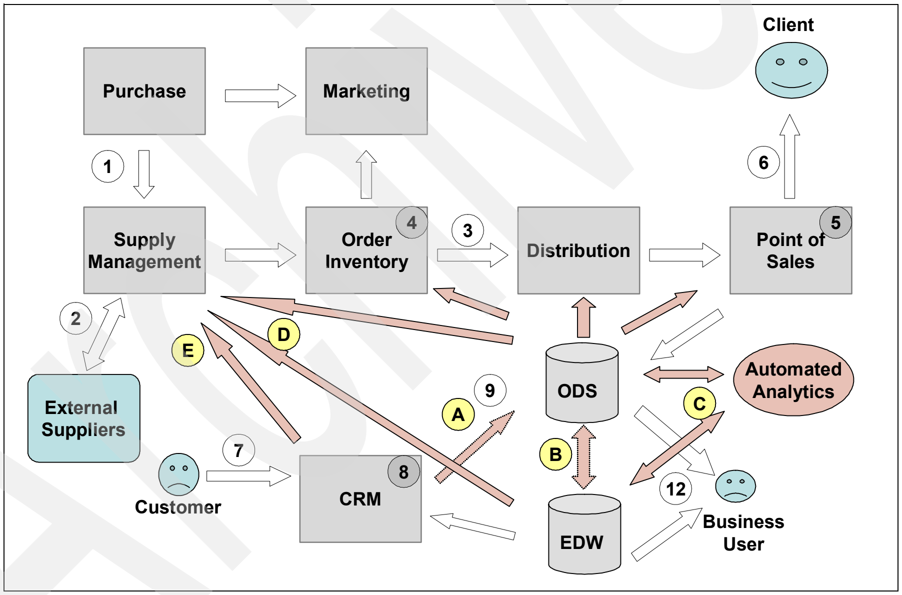


Figure 7-2 The ValueTrend improved data flow

By improving the data flow, and connecting systems that were not connected in the original data flow, much better reaction times and quality of service can be achieved. This is accomplished by using IBM technologies that are discussed in the other sections of this chapter.

The original data flow, combined with an improved real-time enterprise approach, is described in the following list:

1. A purchase order for a ValueTrend hair-dryer is placed by the central purchasing department.
2. The order for the hair-dryer is fulfilled by the supplier and delivered to the ValueTrend central distribution center.
3. The hair-dryer is then distributed to the ValueTrend retail store.
4. The Inventory system is updated to reflect this event.
5. The hair-dryer is now available to satisfy incoming sales orders from the ValueTrend sales channels.
6. Time passes and orders are processed. This should typically result in products being delivered and the result a happy customer.
7. ValueTrend receives calls from a number of unhappy customers complaining about the hair-dryer. They all indicate a belief that it is defective.
8. All returns, and a code indicating the reason for the return, are registered in the CRM system. Records of the returns are created in the CRM system, and are then used as measures for determining customer satisfaction. The reason codes have been implemented to handled all direct customer dialogs. All product return records are given a status code reflecting the seriousness of the defect.
9. The return reason information is collected and transferred to the ODS by a weekly batch ETL process.

    a. This batch oriented process is changed to use low latency data replication that has low performance impact on both the source and target system.

10. The returns information is, at a scheduled point-in-time extracted, transformed, and loaded into the ValueTrend EDW.

    b. Portions of ETL process can either be implemented with data replication or, alternatively, data federation can be used to access the ODS for lookup of current data.

11. The central returns department receives the actual hair-dryers back from the stores, and, with their BI capability, discover that a high percentage of the sold hair-dryers are being returned.

    c. Because the returns information is replicated at low-latency, there is now a fast and easy way to get real-time access to the sales data. There is now benefit from implementing real-time analysis, which can enable automated analytics on our sales and return data.

12. An investigation is launched by quality control. Sales statistics are produced by taking information from EDW and the ODS.

d. When investigating the situation, there is an urgent need to have information about the current situation. Through zero-latency data federation, the systems can be accessed to get a real-time view of the current supply, inventory and sales situations.

13. It is decided by ValueTrend to recall the hair-dryer. Through a press release, and stores notice boards, customers are informed that they can return the ValueTrend hair-dryer and get a full refund.

  e. After this decision is made a fast response can be made and all involved systems informed of the current situation. Using a combination of data federation and EAI for updating the internal systems, there can be direct interaction the external supplier.

In this section we illustrate how low latency can make the output of the analytical processes in the ODS and EDW available to other areas of the operational environment and deliver a real-time operating environment. By using a closed-loop approach we can react to occurrences that are discovered, either manually or automatically in the business intelligence environment.

In this case study we presume that the information needed to make a business decision is available either in the ValueTrend operating environment, or from outside by using integration technologies.

7.2 Tested technologies, results and findings

In this section we get into more detail of the technologies tested during this case study. We not only talk about the technologies, but also discuss some of the findings and result observed. We do not, however, discuss installation and configuration issues in any detail, as these are subjects covered in quite some detail in other redbooks. For more information, see "Related publications" on page 361.

Figure 7-3 depicts the tested technologies and will be used as the base structure through the rest of this chapter to illustrate descriptions of the case study. There were actually two scenarios tested. Test Scenario A used the capabilities of WebSphere Message Broker V5.3. Test Scenario B used the Enterprise Service Bus capability of V6. These two scenarios are defined in more detail throughout this chapter.

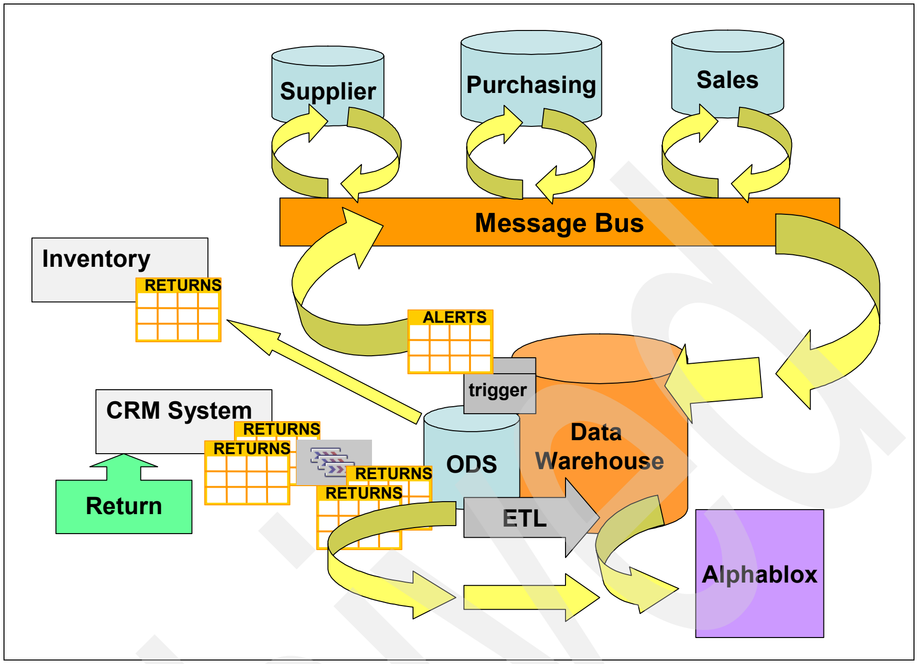


Figure 7-3 Case study scenario A and B data flow

7.2.1 Enterprise Information Integration

The IBM enterprise information integration solution is based on the WebSphere Information Integration (WebSphere II) product family. Under the WebSphere II product portfolio there are various solution areas. These areas are described in more detail in Chapter 4.

Data Federation

Data federation is used to support and access heterogeneous sources of data, using WebSphere II. This data federation is enabled by mechanisms called wrappers. These wrappers contain the required support to access the heterogeneous data sources, and are used by the calling applications. The accessed data is stored on a federated server.

A wrapper performs many tasks, some of which are:

- Connecting to the data source, using the standard connection API of the data source.
- Submitting queries to the data source. For data sources that do not support SQL, one of two actions will occur:

  – For those that support SQL, the query is submitted in SQL.
  – For those that do not support SQL, the query is translated into the native query language of the source or into a series of source API calls.

- Receives results sets from the data source. The wrapper uses the data source standard APIs for receiving results set.
- Responds to federated server queries about the default data type mappings for a data source. The wrapper contains the default type mappings that are used when nicknames are created for a data source object. Data type mappings you create override the default data type mappings. User-defined data type mappings are stored in the global catalog.
- It responds to federated server queries about the default function mappings for a data source. The wrapper contains information required by the federated server to determine if WebSphere II functions are mapped to functions of the data source, and how the functions are mapped. This information is used by the SQL Compiler to determine if the data source is able to perform the query operations. Function mappings created override the default function type mappings, and user-defined function mappings are stored in the global catalog.

The details of installing and configuring the individual wrappers are not covered in this redbook. But there are illustrations of how the WebSphere II configuration looks and works.

As part of the improved workflow, the following data sources and associated WebSphere II wrappers were tested:

1. DB2 UDB
2. IMS via Classic Connect
3. Oracle
4. Microsoft SQL Server
5. Microsoft Excel
6. Table-structured files
7. Web Services
8. XML

The tests were to get and/or push data to and from these different data sources with zero-latency. The interfaces used to get access to these data source were through SQL, JDBC and ODBC connections. DB2 SQL tools, as well as DB2 Alphablox, were used to build ad-hoc reporting environments.

1. DB2 UDB wrapper:

The DB2 UDB DRDA® wrapper was used between various DB2 UDB systems on Linux and Windows, and DB2 on z/OS. We declared a number of nicknames that were used to point to DB2 tables on the remote systems.

The tests performed did push and pull of data to the different DB2 data sources, with no latency.

Some of the data flow and closed-loop tests were performed through the DB2 wrapper to update the orders table on DB2 for z/OS. This was meant to demonstrate how orders could be put on hold and thereby avoid producing and stocking additional, potentially defective, hair-dryers.

Figure 7-4 depicts the configuration of nicknames in the local WebSphere II gateway that are pointing to the tables in DB2 for z/OS. The configuration of these nicknames in accomplished through a simple definition wizard.

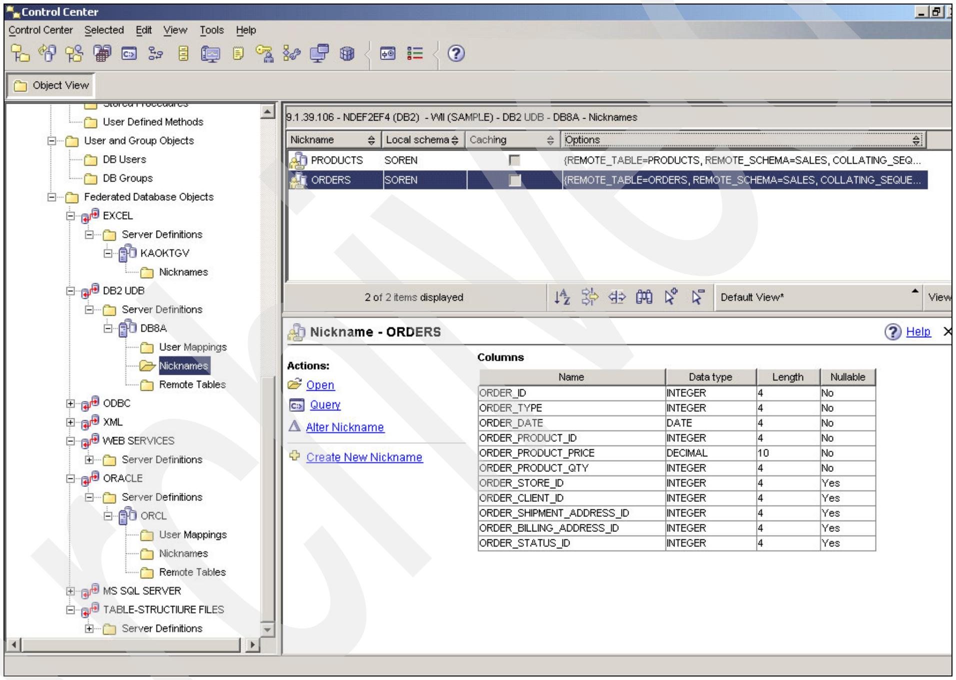


Figure 7-4 DB2 z/OS tables define as nicknames to our WebSphere II gateway

There was also a test performed where the WebSphere II data wrapper used a local nickname to point to a remote nickname used as WebSphere II gateway. This worked very elegantly, providing the option of using the WebSphere II gateway to hide any complexity of non-DB2 sources and make them look to distributed DB2 users as though they are a local DB2 source.

2. **IMS via Classic Federation for z/OS wrapper:**

This product provides direct, real-time SQL access to mainframe databases and files without mainframe programming. JDBC or ODBC SQL statements are dynamically translated into native read/write APIs.

It can access data stored in VSAM, IMS, CA-IDMS, CA-Datacom, Software AG Adabas, and DB2 UDB for z/OS databases. However, in the case study only and IBM IMS environment on z/OS was used. It is depicted in Figure 7-5.

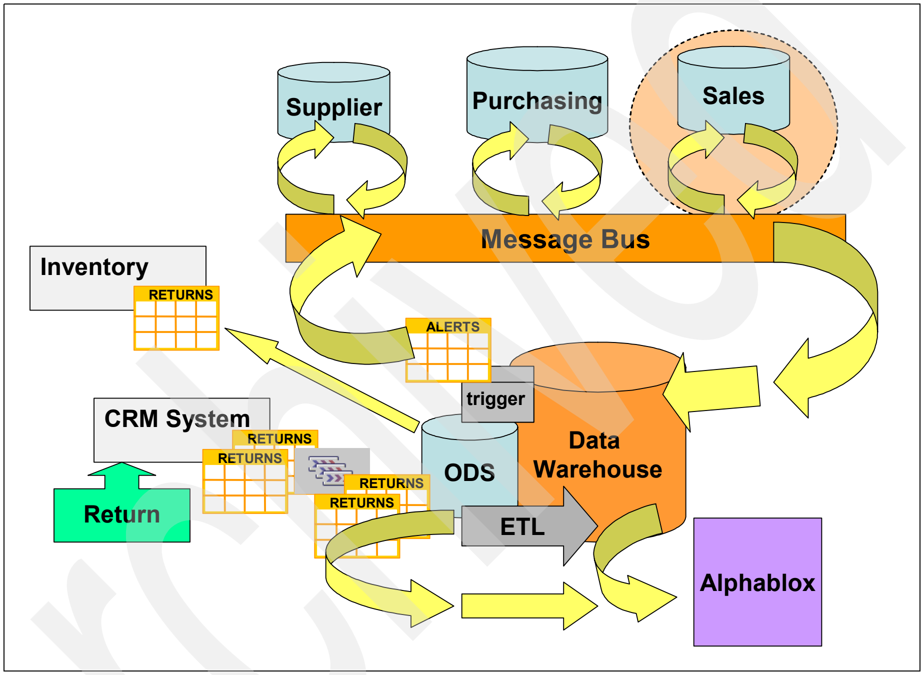


Figure 7-5 IMS WebSphere II gateway solution

In 7.2.5 "Test with combined EII and ETL" on page 340,there is a more detailed description of this test. With the combination of EII and ETL test a direct zero-latency connect to a IMS system was used to combined with a Web services wrapper to connect to WebSphere DataStage.

The Classic Connect ODBC enables the IMS data sets, and makes them available to the WebSphere II gateway solution. Nicknames are then defined to access the IMS data, as shown in Figure 7-6.

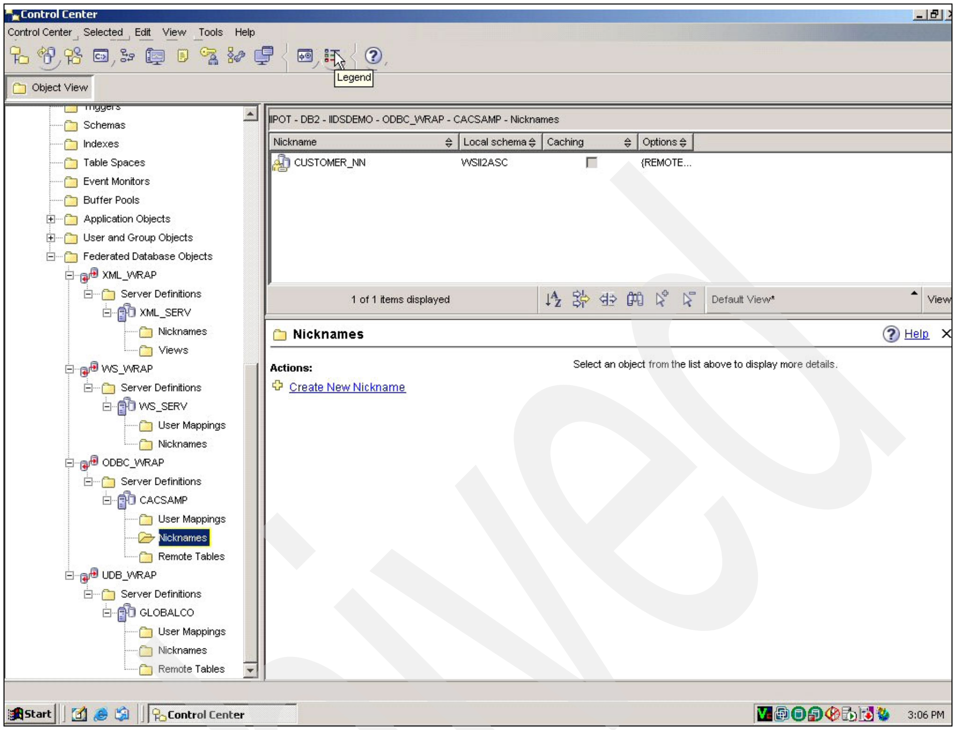


Figure 7-6 Definition ODBC wrapper that connects to IMS z/OS

3. Oracle wrapper:

Setting up and testing the Oracle client configuration file is part of the task of adding Oracle data sources to the WebSphere II federated servers.

The Oracle wrapper was setup and used to provide access to Oracle via the WebSphere II gateway. It is also part of the setup with one table to illustrate access to Oracle as part of an overall closed-loop test. This is depicted in Figure 7-7.

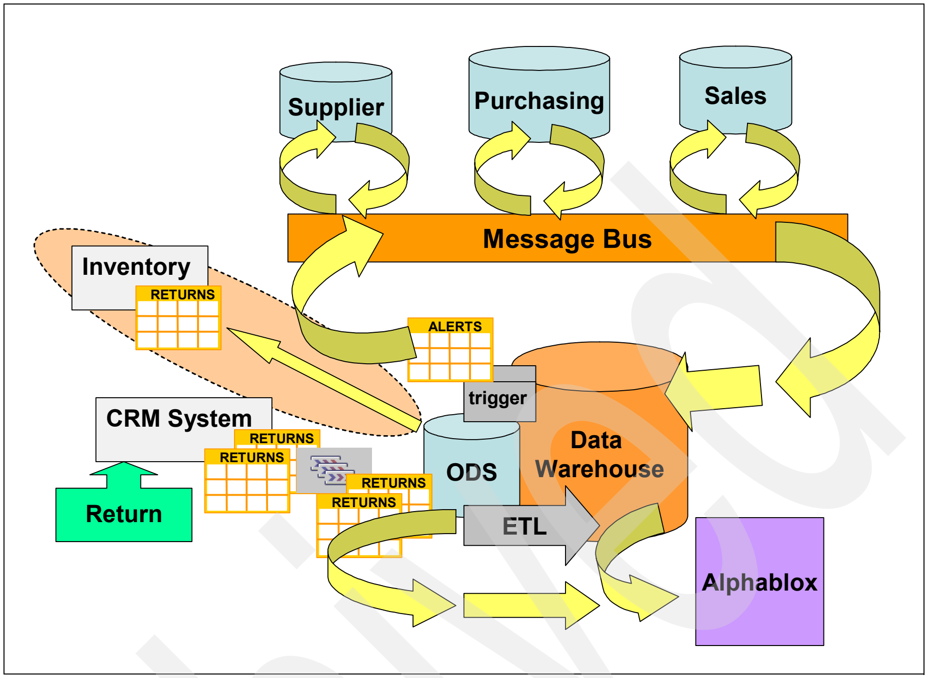


Figure 7-7 Oracle WebSphere II gateway solution

The client configuration file is used to connect to Oracle databases, using the client libraries that are installed on the WebSphere II federated server. This file specifies the location of each Oracle database server and type of connection (protocol) for the database server.

Nicknames are defined to access the Oracle database, as shown in Figure 7-8.

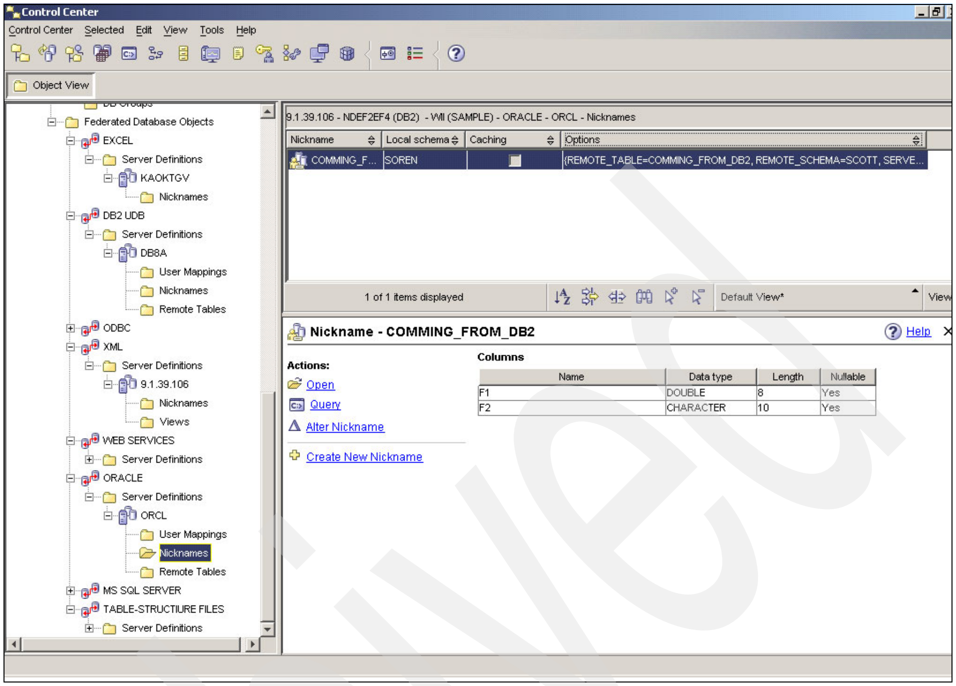


Figure 7-8 WebSphere II nicknames to Oracle table definition

The test was to set up Oracle on a Windows server and define a table in Oracle. Then a nickname was defined in the WebSphere II gateway, and finally a DB2 server defined a nickname that points to the WebSphere II gateway nickname. This scenario worked very well and is illustrated in Figure 7-9.

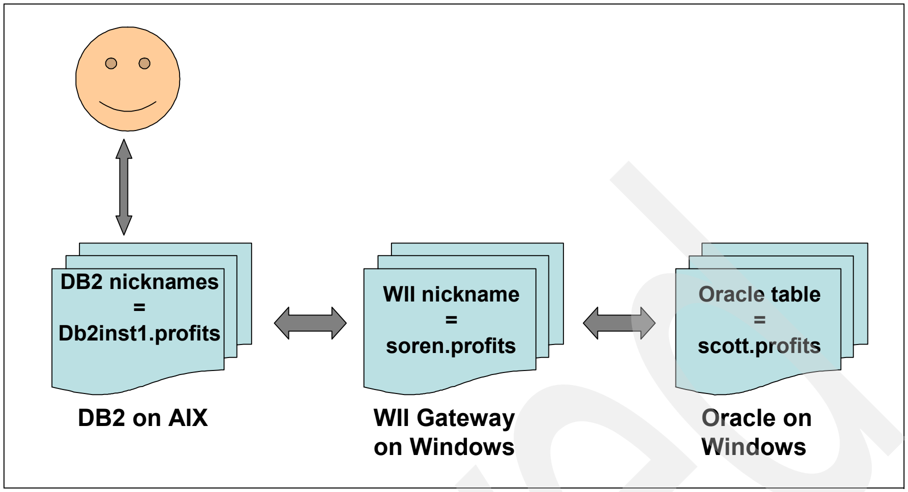


Figure 7-9 Test of a DB2 nickname to WebSphere II nickname to Oracle table

4. Microsoft SQL Server wrapper:

The MS SQL server wrapper was tested the same as the Oracle wrapper. However, in addition, this wrapper was available to validate the overall relational connectivity offering in the closed-loop scenario.

Configuring the federated server to access Microsoft SQL Server data sources involves supplying the federated server with information about the Microsoft SQL Server data sources and objects to be accessed. This setup was very similar to the setup of the Oracle wrapper. The SQL Server wrapper used in the case study is depicted in Figure 7-10.

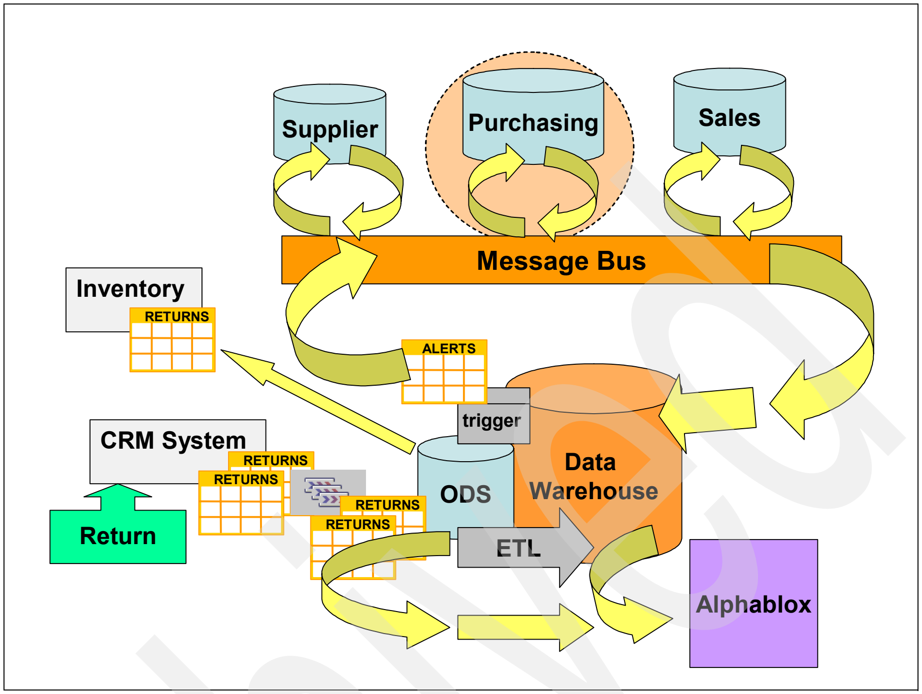


Figure 7-10 SQL Server WebSphere II gateway solution

5. **Microsoft Excel wrapper:**

Much of the data today often ends up in spreadsheets. These are used as front-ends and analysis tools to manipulate and sometimes summarize data. The Excel wrapper gives the opportunity to include these spreadsheets in the overall reporting environment and access the spreadsheets through the same interface and with the same tools as any other data source in the enterprise.

6. **Table-structured files wrapper:**

A table-structured file has a regular structure consisting of a series of records, where each record contains the same number of fields, separated by an arbitrary delimiter.

Using the module called a wrapper, WebSphere Information Integrator can process SQL statements that query data in a table-structured file as though it were contained in an ordinary relational table or view. As for the other wrappers, this enables data in a table-structured file to be joined with relational data or data in other table-structured files.

During the case study we used the table-structure files wrapper to simulate the reading and creation of files that should be made available for older or earlier applications.

7. Web services wrapper:

Web services providers are described using the Web Services Description Language (WSDL). The Web services wrapper can access Web service providers. The Web service provider implements a service and publishes the interface to a service broker, such as UDDI. The service requester can then use the service broker to find a Web service. When the requester finds a service, the requester binds to the service provider so that the requester can use the Web service. The requester invokes the service by exchanging SOAP (simple object access protocol) messages between the requester and provider.

The SOAP specification defines the layout of an XML-based message, which is contained in a SOAP envelope. The envelope consists of an optional SOAP header and a mandatory SOAP body. The SOAP header can contain information about the message, such as encryption information or authentication information, and the SOAP body contains the message. The SOAP specification also defines a default encoding for programming language bindings, which is called the SOAP encoding.

The key to the Web service is the WSDL document. The WSDL document is an XML document that describes Web services in terms of the messages that it sends and receives. Messages are described by using a type system, which is typically the XML schema. A Web service operation associates a message exchange pattern with one or more messages. A message exchange pattern identifies the sequence and cardinality of messages that are sent or received, as well as who the messages are logically sent to or received from. An interface groups together operations without any commitment to the transport or wire format. A WSDL binding specifies transport and wire format details for one or more interfaces. An endpoint associates a network address with a binding. A service groups together endpoints that implement a common interface. The messages can contain document-oriented information or process-oriented information, which is also known as remote procedure calls (RPC). A WSDL document can contain one or more Web services.

The case study tests used the WebSphere II Web services wrapper to access external Web services on the internet. It was used to access a simple Web service that provided a real-time currency exchange functionality between to currencies. In addition, a Web service that could make real-time price checking comparisons was considered. These services could be used to look at average sales market prices for specific products and help our retailer to make better real-time competitive price analysis and adjustments

Using the Web service wrapper functionality in WebSphere II enables easy incorporation of external data and application logic through the Web service interface layers.

8. **XML wrapper:**

With the XML wrapper, XML data can be mapped from an external data source into a relational schema that is composed of a set of nicknames. The structure of an XML document is logically equivalent to a relational schema in which the nested and repeating elements are modeled as separate tables with foreign keys.

The nicknames that correspond to an XML document are organized into a tree structure in which the child nicknames map to elements that are nested within the element that corresponds to the parent nickname. When nested elements are repeated or have distinct identities with complex structures, separate nicknames can be provided for each nested element. Child and parent nicknames are connected by primary and foreign keys that are generated by the wrapper.

XPath expressions are used to map an XML document into a relational schema that is composed of a set of nicknames. XPath is an addressing mechanism for identifying the parts of an XML file (for example, the groups of nodes and attributes within an XML document tree). The basic XPath syntax is similar to file system addressing.

Each nickname is defined by an XPath expression that identifies the XML elements representing individual tuples, and a set of XPath expressions that specifies how to extract the column values from each element. Similar to the Table-structured wrapper, the XML wrapper was used to simulate how XML files that are used for data transfer between systems can be created and used, either for external or internal systems.

Data Replication

There are differing IBM replication technologies available to help determine the best match for the simple low-latency requirements of data movement.

Figure 7-11 is a picture of the startup screen in the Replication Center Launchpad. The option is available to move to any of the WebSphere II supported kinds of replication, Q-replication, event publishing and SQL-replication. In the case study test scenarios only Q-replication and event publishing were used.

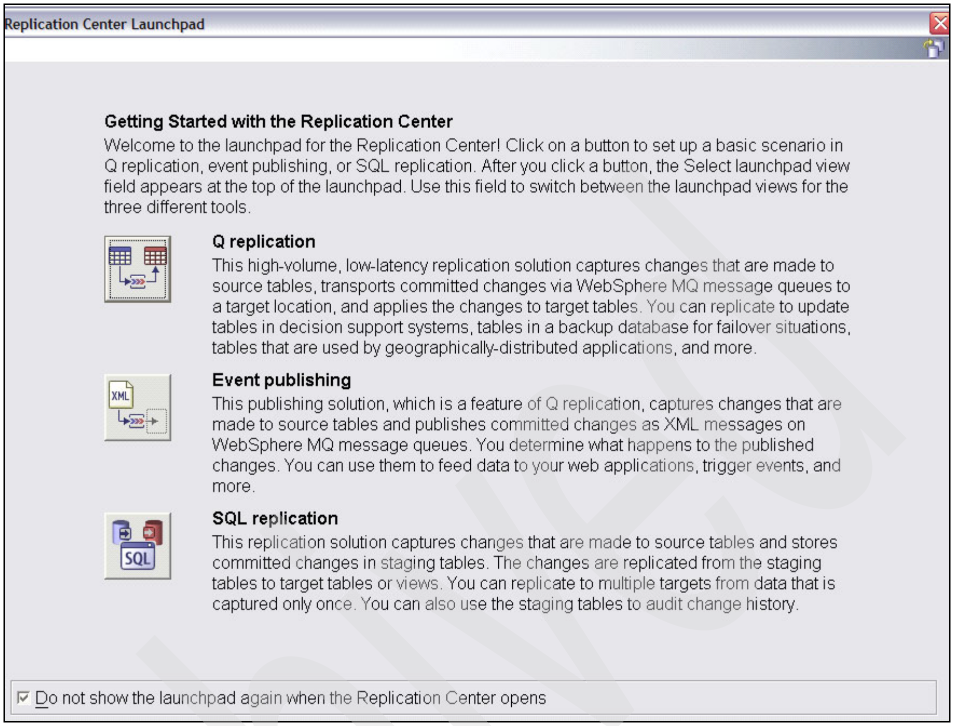


Figure 7-11 Replication Center launchpad

Data replication and SQL replication

Although no testing was performed with SQL-replication, the SQL-replication launchpad in Figure 7-12 is used to show the differences in the replication architectures.

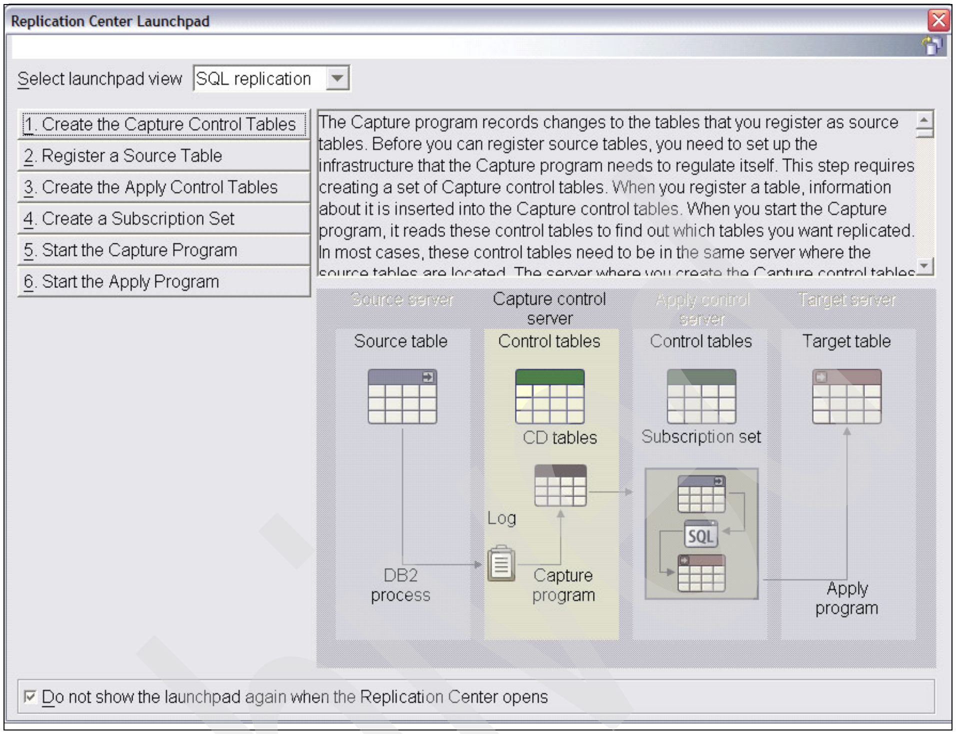


Figure 7-12 SQL-replication launchpad

Data replication and Q-replication

As describe in Chapter 4, “IBM technologies supporting real-time” on page 111, the IBM implementation of data replication is a subset of functionality under the WebSphere Information Integrator framework. A number of tests have been performed in this redbook with data replication. However, the tests used in the case study have primarily been performed with the Q-replication and event publishing features of WebSphere Information Integrator.

Traditionally, replication has been used to consolidate or distribute data, create query ready copies of OLTP data, and enable maintenance processes for data warehousing. But new uses of database replication technology have emerged in the last few years. Some customers want to use replication for traditional purposes, but need the changed-data feed to be real time rather than a scheduled batch activity.

A smaller but growing set of solutions for customers is creating geographically dispersed applications that require multiple update ready database copies. Each of these replication scenarios depends on the ability to replicate data reliably, with low latency, and, in some cases, multidirectional, with provisions for robust conflict detection and resolution. Q-replication was selected to satisfy the case study requirements for achieving a lower latency when moving data between the CRM system and the ODS. It is depicted in Figure 7-13.

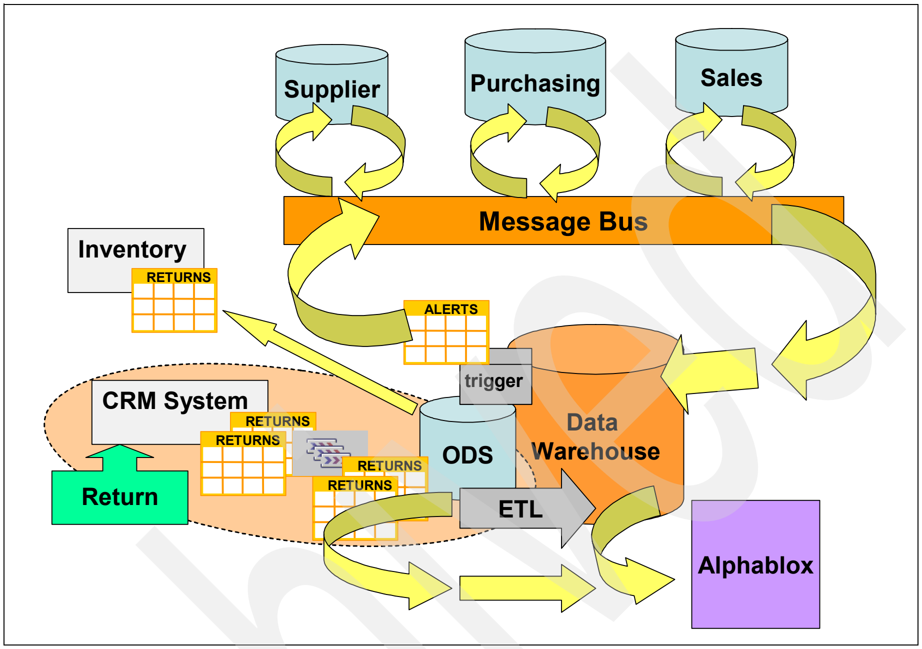


Figure 7-13 Q-replication

This section is an introduction to the replication setup for the case study test environment. Figure 7-14 depicts the program used to help generate the configuration scripts for setting up Q-replication.

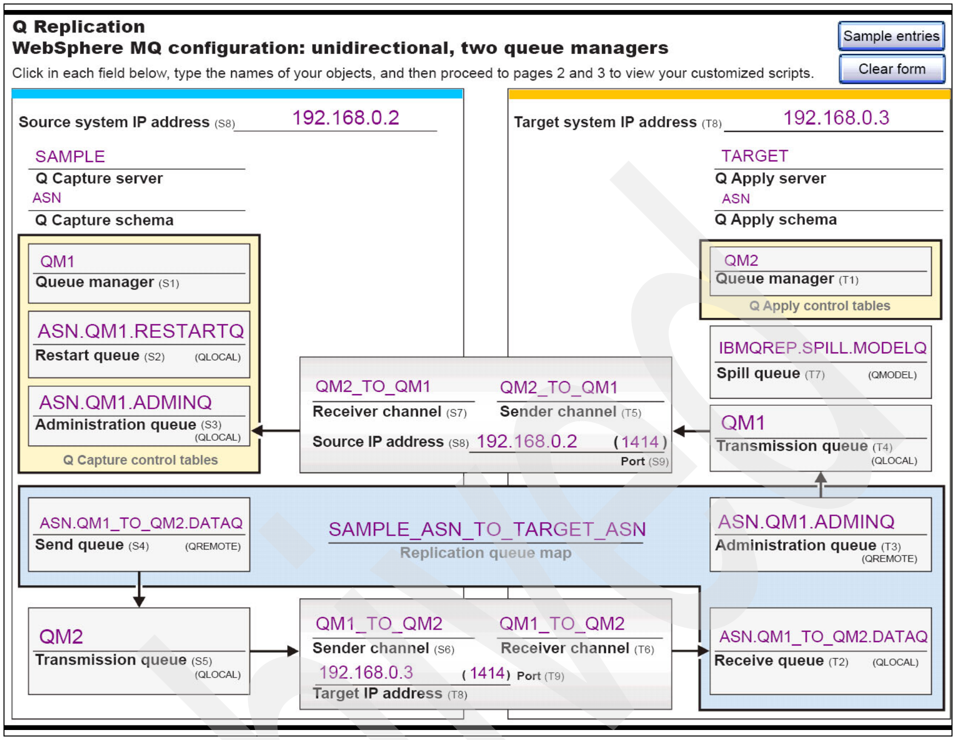


Figure 7-14 Q-replication setup of MQ configuration input screen

An example of the source system output from the MQ configuration setup is shown in Figure 7-15.

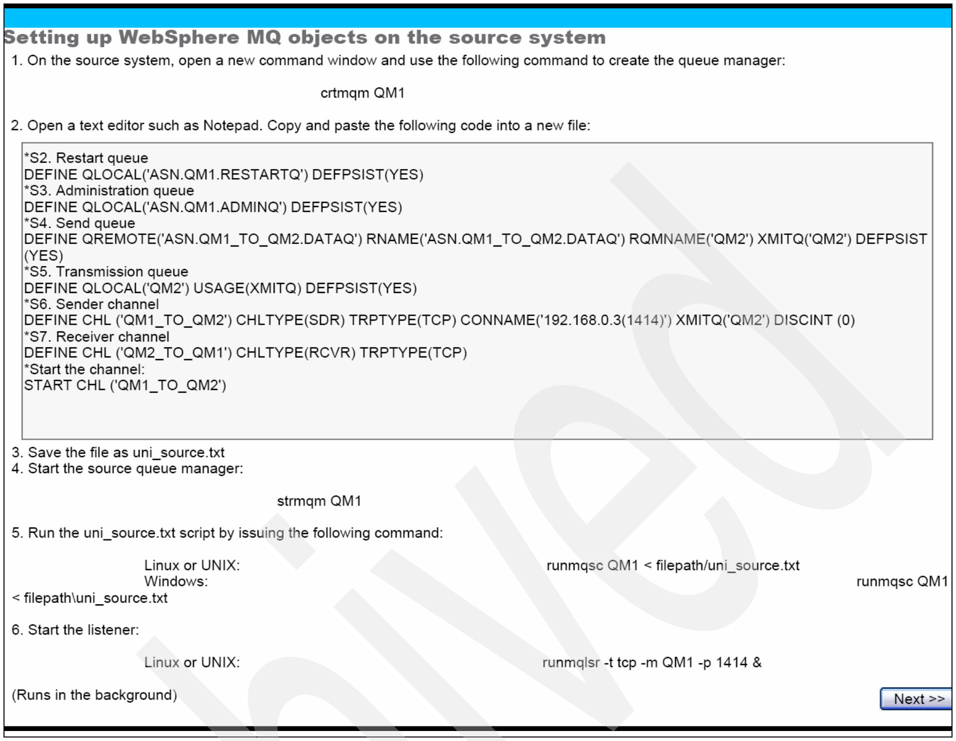


Figure 7-15 MQ configuration script for the source system

We have chosen to use an image of the Q-replication launchpad, shown in Figure 7-16 to depict the Q-replication architecture.

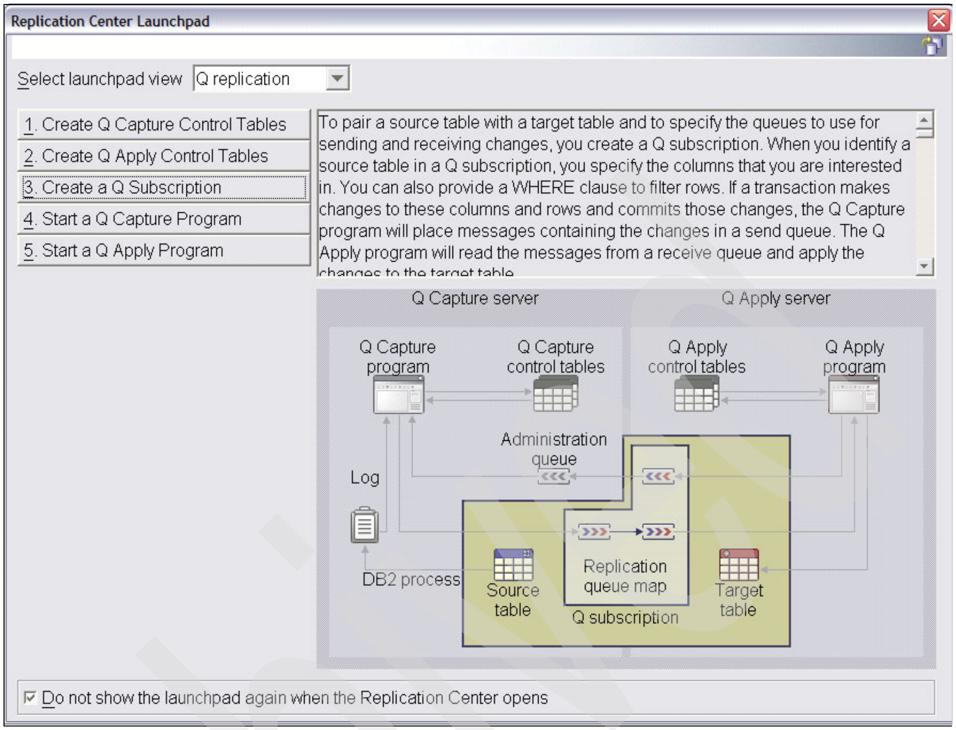


Figure 7-16 Q-replication launchpad

Replication Center was used to manage and control the environment. An example of this setup is shown in Figure 7-17.

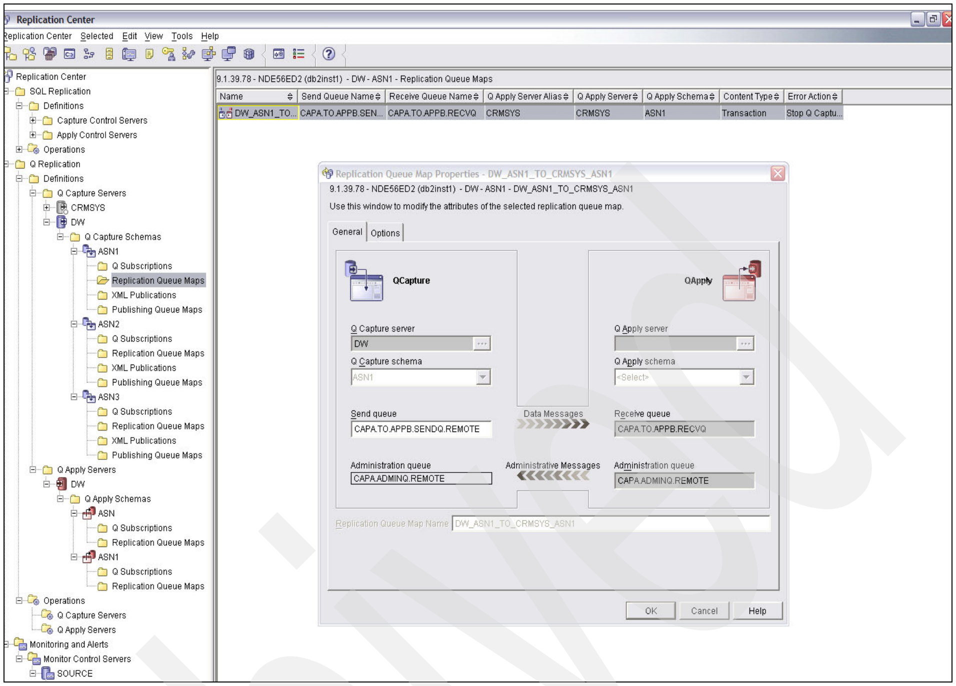


Figure 7-17 Sample screen of a Q-replication setup

Even though there were no in depth measurements of the latency profiles observed when manipulating the latency parameters of the Q-replication configuration, sub-second latency results were observed. As part of the test, hundreds of records per second were sent through the replication setup. This was also successful even though there had been no configuration adjustments made to support larger scales of throughputs.

The architecture of Q-replication does require some installation and configuration efforts. However when installed and configured, excellent performance with very low resource usages was observed.

Event Publishing

As with data replication, IBM has implemented event publishing within the WebSphere Information Integrator framework.

Event publishing was used in this redbook case study, and is depicted in Figure 7-18.

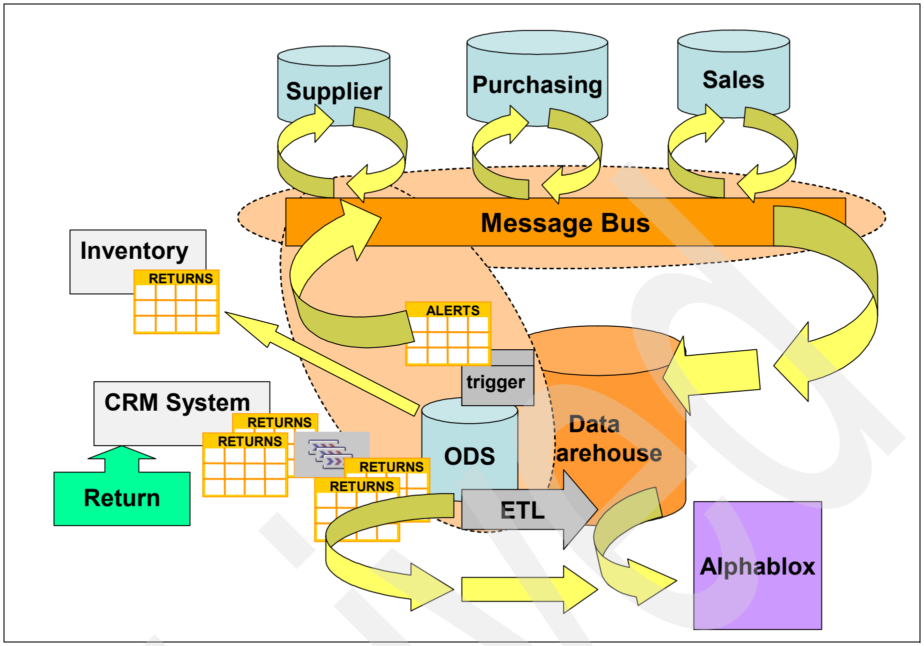


Figure 7-18 WebSphere II Event Publishing

Changes to data can be published as individual row messages or as messages representing all the changes within a transaction. Any application or service that integrates either with WebSphere MQ directly or supports Java Message Service (JMS) can asynchronously receive the data changes as they occur. This enables creation of data-driven workflow processing. For example, a change in an inventory value could be used to drive a product restocking workflow, or the addition of a new customer could initiate a welcome e-mail, credit verification, and accounting updates. This creates an application-independent, loosely-coupled integration that is more adaptable to changing application environments. In the above inventory example, while multiple applications may impact the value of the inventory level, a single point of integration (the inventory data items themselves) drives the workflow. Changes to the applications that impact the inventory level can be made with no impact on the event driven integration. Other members of the WebSphere Information Integrator product family can be used to capture data changes made to DB2 UDB for z/OS or IMS databases, as well as changes made to VSAM files through CICS as shown in Figure 7-19.

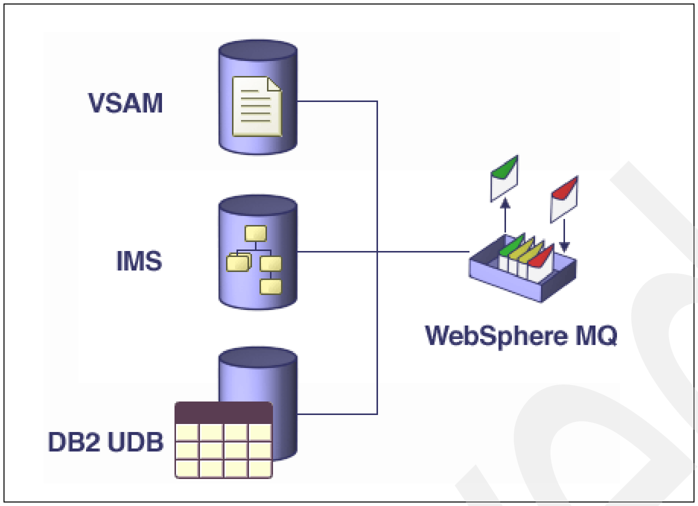


Figure 7-19 WebSphere II Event Publishing overview

These changes are published to WebSphere MQ message queues using the same XML message format.

Using WebSphere II Event Publisher, WebSphere Message Broker can receive changes as they occur from a DB2 UDB database and, for example, automatically update an SAP application.

It is possible to do some filtering of database changes such that only the data that the receiving application is interested in are published.

In Figure 7-20 we show the Event Publish launchpad, which illustrates the architecture of Event Publishing and the steps required to setup this environment.

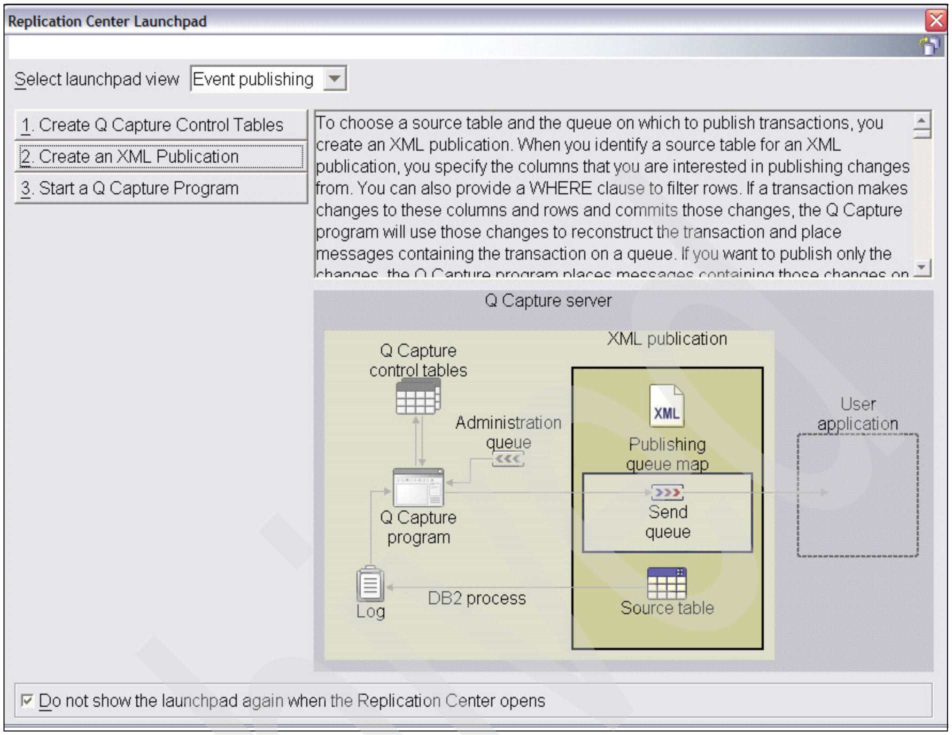


Figure 7-20 Event Publishing launchpad

Figure 7-21 depicts an example of the Replication Center - Event Publisher setup and maintenance facilities. What is shown here is an example of the replication queue maps.

In event publishing, a publishing queue map identifies the WebSphere MQ queue that a Q-Capture program uses to transport data and communicate with the user application that is receiving the XML messages. Each publishing queue map identifies one send queue, which is the WebSphere MQ queue where the Q-Capture program sends source data and informational messages.

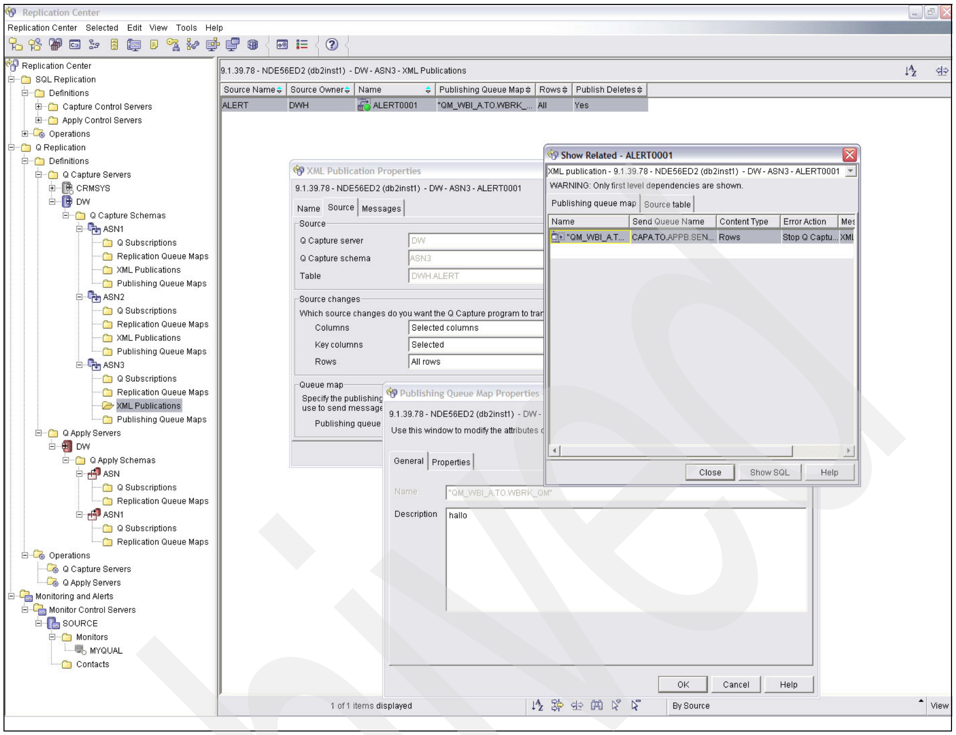


Figure 7-21 Sample screen from Event Publishing setup

In Figure 7-8 on page 294 is an example of an XML message that is put on the receiver queue from WebSphere II Event Publisher.

In Test Scenario A, the WebSphere II Event Publisher was used to capture events in the ODS indicating when the decision to withdrawn the hair-dryer from the market was made. Initially this test was performed with a trigger updating the ALERT table. Event Publisher captures this event with sub-second latency and Event Publisher creates the XML message and places it on the event queue. Then, as described in “Enterprise Application Integration” on page 324, this MQ message queue is read, and the WebSphere environment processes this message and informs the supplier, through Web-services, of the new situation.

In Test Scenario B built-in DB2 functions were used to send a SOAP envelope to the WebSphere ESB message flow, using triggers. Test Scenario A and Test Scenario B are depicted on Figure 7-22.

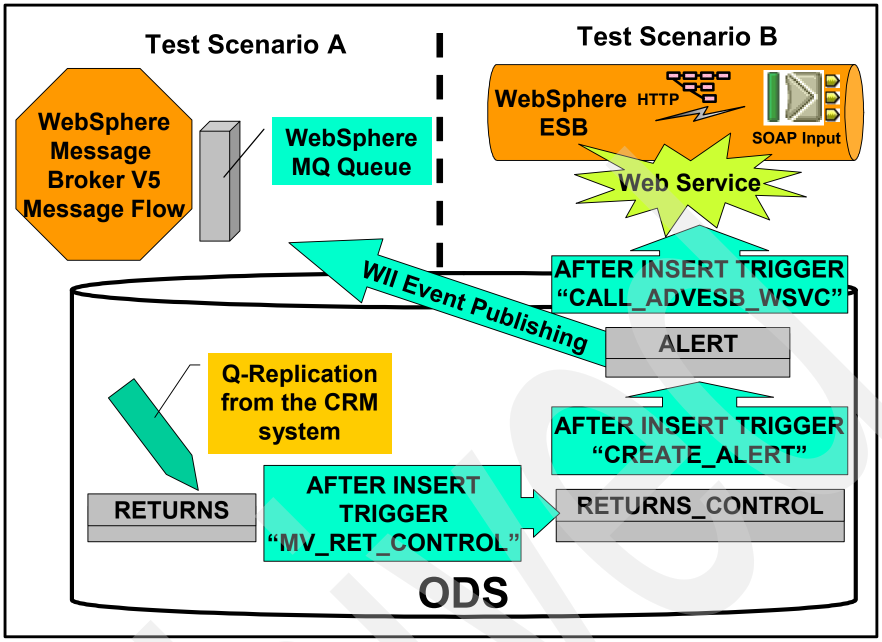


Figure 7-22 Test Scenario A and Test Scenario B

Basically, the control returns flow inside the ODS is the same for both scenarios. It is when the alert table is replicated to activate other applications in the enterprise where a difference can noticed. In Test Scenario A, WebSphere II reads the log and for every insert and generates an event in an MQ queue. In Test Scenario B, built-in functions are used to call a Web service and initiate the WebSphere ESB message flow.

Triggers

Triggers were used in the test scenario to close the loop inside the ODS, as depicted in Figure 7-23.

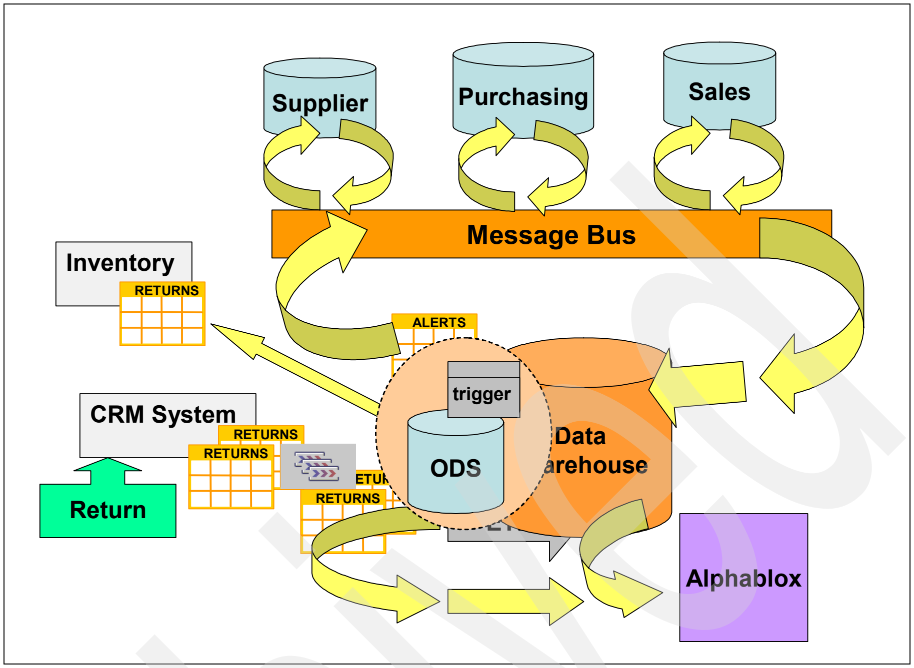


Figure 7-23 Triggers in our test scenario

Example 7-1 shows the trigger definition code used to capture inserts in the returns table in the ODS. The trigger is a simple insert trigger that performs a simple validation regarding the severity of the return type.

Example 7-1 Trigger definition to create in returns control table

```
CREATE TRIGGER DWH.MV_RET_CONTROL
AFTER INSERT ON DWH.RETURNS
REFERENCING NEW AS N
FOR EACH ROW
WHEN ( N.RET_TYPE_ID = 1 )
INSERT INTO DWH.RETURNS_CONTROL(RET_PD_ID, RET_ID) VALUES (N.RET_PD_ID,
N.RET_ID)
```

Example 7-2 shows the trigger definition used to create an alert when predefined thresholds have been met. The thresholds and more advanced analysis could easily be applied here. This event could also trigger a series data mining routines that would handle far more advanced decision support regarding when to create an alert.

Example 7-2 Trigger to create an alert

```
CREATE TRIGGER DWH.CREATE_ALERT
AFTER INSERT ON DWH.RETURNS_CONTROL
REFERENCING NEW AS N
FOR EACH ROW
WHEN((SELECT COUNT(RET_ID) from dwh.RETURNS_CONTROL
where RET_PD_ID=N.RET_PD_ID) = (SELECT MAX_RETURNS from DWH.MAX_RETURNS))
INSERT INTO DWH.ALERT(PD_ID, NUMBER_RETURNS, DESCRIPTION, TIME) VALUES
(N.RET_PD_ID, (SELECT COUNT(RET_PD_ID) from dwh.RETURNS_CONTROL where
RET_PD_ID=N.RET_PD_ID), 'THIS PRODUCT IS WITH A HIGH NUMBER OF RETURNS, SHOULD
BE INVESTIGATED', CURRENT TIMESTAMP)
```

Example 7-3 shows the trigger definition used to clean the returns control table after an alert has been raised.

Example 7-3 Trigger definition to clean returns control table

```
CREATE TRIGGER DWH.CLEAN_RETURNS_CTL
AFTER INSERT ON DWH.RETURNS_CONTROL
REFERENCING NEW AS N
FOR EACH ROW
WHEN((SELECT COUNT(RET_ID) from dwh.RETURNS_CONTROL where
RET_PD_ID=N.RET_PD_ID) > (SELECT MAX_RETURNS from DWH.MAX_RETURNS))
DELETE FROM DWH.RETURNS_CONTROL WHERE RET_PD_ID=N.RET_PD_ID
```

Example 7-4 was uniquely used in Test Scenario B and shows the trigger definition used to call the SOAP function that calls the WebSphere ESB message flow Web service.

Example 7-4 Trigger definition to call the WebSphere ESB message flow Web Service

```
CREATE TRIGGER DWH.CALL_ADVESB_WSVC
AFTER INSERT ON DWH.RETURNS_CONTROL
REFERENCING NEW AS N
FOR EACH ROW
VALUES (call_Svce_High_Ret(N.RET_PD_ID,(SELECT COUNT(RET_PD_ID) from
dwh.RETURNS_CONTROL where RET_PD_ID=N.RET_PD_ID),'THIS PRODUCT IS WITH A HIGH
NUMBER OF RETURNS, SHOULD BE INVESTIGATED'));
```

The triggers were used to illustrate both of the closed-loop test scenarios. By providing a low-latency setup of this integration, we showed in Test Scenario A that on its own, the Event Publisher is a good choice when the desire is to capture changes in DB2 and widely distribute these events in an open standard aimed for non-DB2 environments.

In many environments, the triggers could likely be much more complex in their implementation, and other more comprehensive decision support tools are often used to deliver the similar functionality.

There is, of course, an overhead in doing this type of Event Publishing, when compared to doing it through either data replication or with ETL type technologies. The case study tests did not look at large numbers of events published through the Event Publisher, but were concentrating the tests around the capabilities and the openness of the architecture. With large data volumes, traditional data provision and transformation technologies should be evaluated as well as WebSphere Event Publisher.

Regarding Test Scenario B, the function created to call the WebSphere ESB message flow uses three DB2 functions: XML2CLOB, XMLELEMENT and DB2XML.SOAPHTTPV.

- XML2CLOB returns the argument as a CLOB value. The schema is SYSIBM. The argument must be an expression of data type XML, and the result has the CLOB data type.
- XMLELEMENT constructs an XML element from the arguments. The schema is SYSIBM. The function name cannot be specified as a qualified name. This function takes an element name, an optional collection of namespace declarations, an optional collection of attributes, and zero or more arguments that will make up the element's content. If the element content is null, the result will be an empty element. The result data type is XML.
- DB2XML.SOAPHTTPV. DB2 Universal Database and Web services functions can be extended with the ability to invoke Web services from within Structured Query Language (SQL) statements. This is done by invoking a set of user-defined functions (UDFs) that provide a high-speed client simple object access protocol (SOAP) over Hypertext Transfer Protocol (HTTP) interface to accessible Web services. These functions can be called directly from SQL statements.

The SOAP body can be constructed according to the Web services description language (WSDL) of a Web service. You can also use the Web service User-Defined Function (UDF) tool in WebSphere(R) Studio Application Developer (WSAD) to automatically generate specific UDFs. These UDFs can invoke operations that are defined by a user-specified Web services description language file. The generated UDFs are DB2 UDB functions that do the following:

- Provide the parameters for the Web service request.
- Invoke the SOAP client functions.
- Map the result of the Web service invocation to the return types specified by the user.

The function created was called call_Svce_High_Ret and is depicted in Example 7-5.

Example 7-5 call_Svce_High_Ret description

```
CREATE FUNCTION call_Svce_High_Ret (number_of_returns INTEGER, product_id
INTEGER, description VARCHAR(200))
  RETURNS VARCHAR(40)
  LANGUAGE SQL READS SQL DATA
  EXTERNAL ACTION NOT DETERMINISTIC
  RETURN
    WITH
--1. Perform type conversions and prepare SQL input parameters for SOAP
envelope
      soap_input (in)
        AS
        (VALUES VARCHAR(XML2CLOB(XMLELEMENT(NAME "return",
                        XMLELEMENT(NAME "number_of_returns",
number_of_returns),
                        XMLELEMENT(NAME "product_id", product_id),
                        XMLELEMENT(NAME "description", description))))),

--2. Submit SOAP request with input parameter and receive SOAP response
      soap_output (out)
              AS
     (VALUES DB2XML.SOAPHTTPV('http://9.43.86.102:7080/WSHOST1','',(SELECT in
FROM soap_input)))

--3. Shred SOAP response and perform type conversions to get SQL output
parameters

    SELECT out FROM soap_output;
```

7.2.2 Extract, transformation, and load

In this section WebSphere DataStage is used to provision data from the CRM application server to the operational data store. The data is pulled from the data source and pushed into the ODS on a weekly schedule.

WebSphere DataStage

This section describes extract, transform, and load process for data about previously purchased products that have been returned to ValueTrend. Previously purchased products may be returned by customers for many different reasons. For the specific scenario described in 7.1.2 “The data flow” on page 283, the product returned has a defect that could pose a severe safety hazard. However, for the actual ETL process this detail is irrelevant.

WebSphere DataStage, as used in the test scenario, is depicted in Figure 7-24.

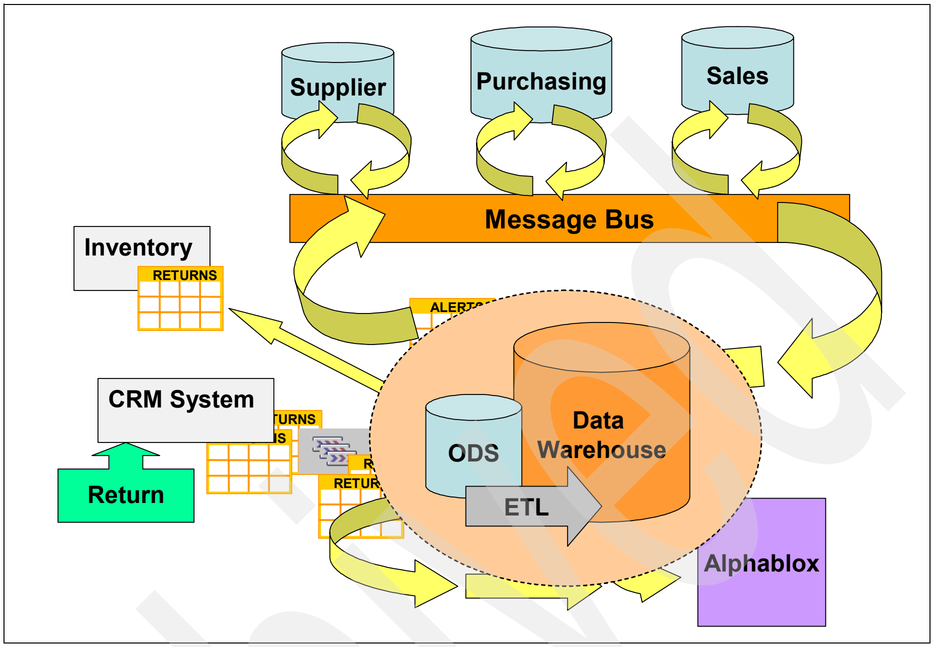


Figure 7-24 DataStage in the test scenario

Figure 7-2 illustrated that the data on returned items is first entered into the ValueTrend CRM system, which is running in a Linux Red Hat environment with DB2 UDB Version 8.2 as the data repository. The CRM system is a heavily loaded system, as all divisions of ValueTrend uses this system to maintain and register all customer engagements and activities. For these reasons the batch windows in which the work is performed has diminished over time and thereby has limited the ETL workload windows to only run on a weekly basis.

To achieve the data movement task, WebSphere DataStage was used. This product was formerly known as Ascential® DataStage®. Actually, in this test scenario, we used an early version of DataStage Designer (version 7.5.1.A) which, as you will notice, still has the Ascential logo on the screens. Also notice that, as depicted in Figure 7-25, the DataStage project is called ValueTrend.

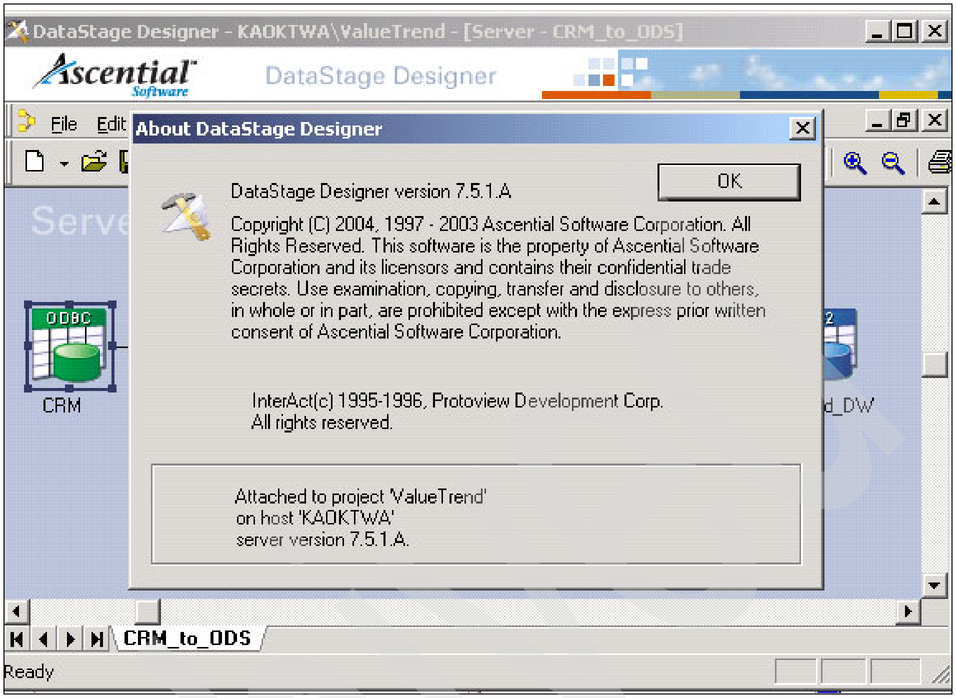


Figure 7-25 WebSphere DataStage 7.5.1

Refer again to Figure 7-2. The target ODS/Data Warehouse box is in the center of the diagram. The data warehouse environment runs on AIX 5.2, and also uses DB2 Version 8.2.

To prepare the ETL process for the data on returned items, an ODBC data source has been defined for the CRM system. The next step created the DataStage server job to connect to the ODBC data source, and pull the data from the CRM system. Since the data is being moved into the ODS, there is no requirement to manipulate any transaction in the transformation step. The transformation simply maps the source table columns to the target table columns and then inserts the data in the operational data store.

Figure 7-26 shows the DataStage job, *CRM_to_ODS,* and how it has been scheduled to run once a week using the scheduling function of DataStage Director.

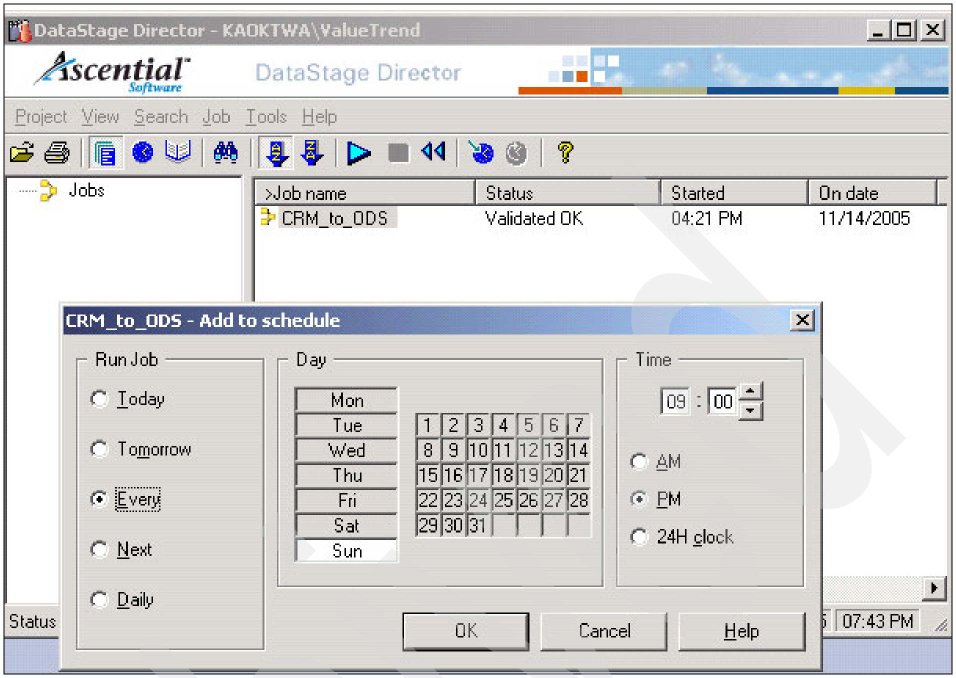


Figure 7-26 DataStage Director scheduler

As can bee seen, the data movement from the CRM system to the ODS system is fairly straight forward. The main objective is to illustrate the business case and how ValueTrend evolves towards the real-time enterprise by changing IT and organizational processes.

7.2.3 The dashboard

At a high level, within the on demand environment, WebSphere Portal provides collaboration and access capabilities. It can be used to improve communication and collaboration among employees, customers, partners, and suppliers. WebSphere Portal provides dynamic access to information and applications, which allows businesses to respond in real time to current conditions and information. These capabilities, in turn, improve the efficiency and effectiveness of businesses.

The dashboard used in this test scenario is depicted in Figure 7-27.

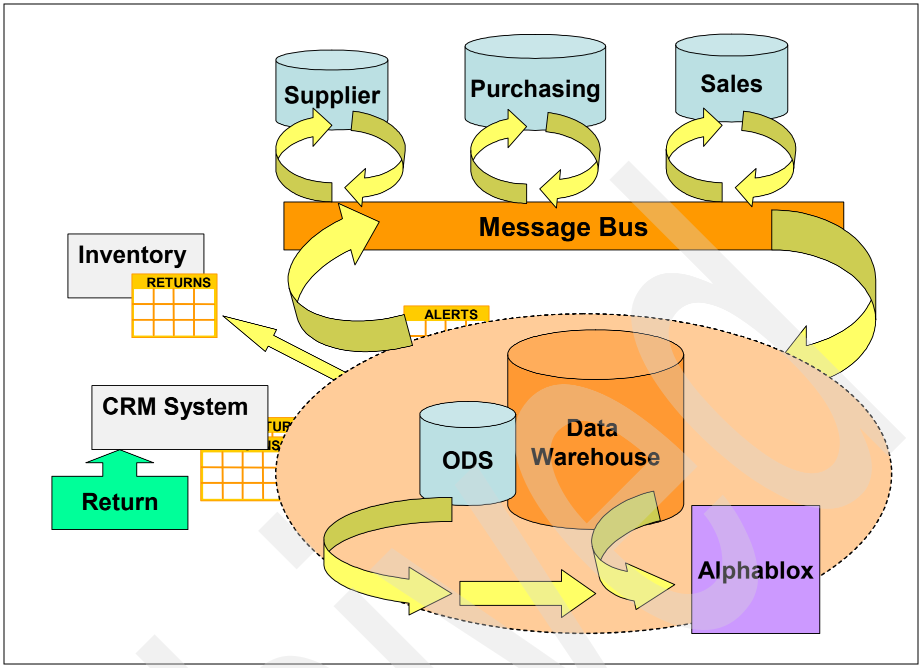


Figure 7-27 Test scenario dashboard

At a more technical level, WebSphere Portal runs as a Web application on IBM WebSphere Application Server. Portals (which are Web applications) can access these services through standards-based APIs, including out-of-the-box portlets, and other portlets that can be downloaded. Builders and tools are also available on WebSphere Portal and Lotus Workplace Catalog. Figure 7-28 shows a high level view of the WebSphere Portal architecture.

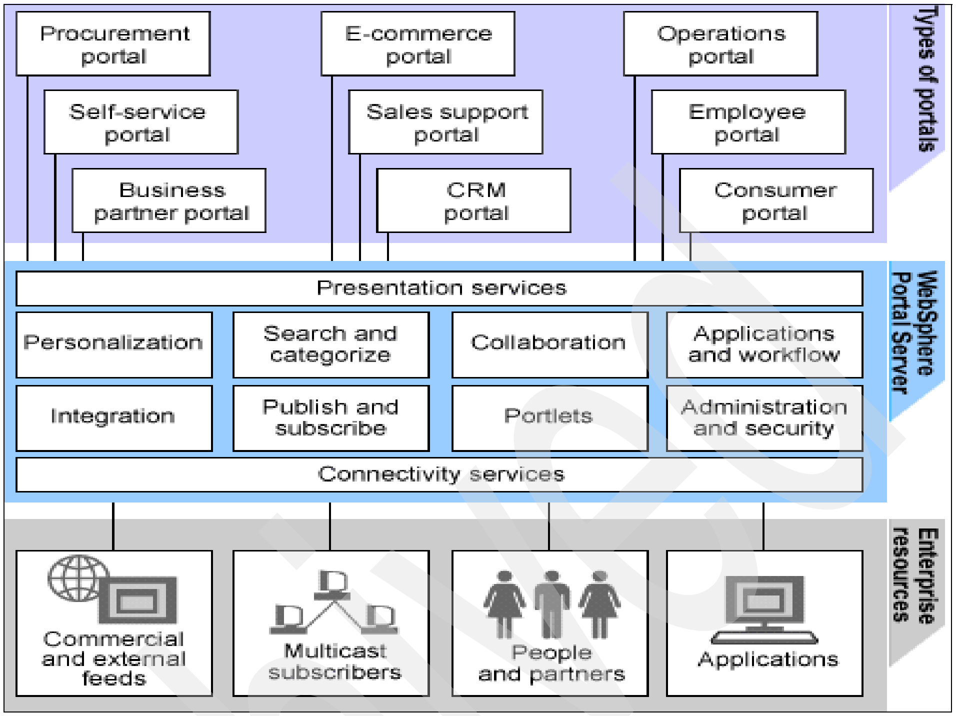


Figure 7-28 Services for all types of portals

In the case study, Portal was used to host the Alphablox portlet and also to integrate it with the CRM system, simulating an operational Call Center application. This is depicted in Figure 7-29.

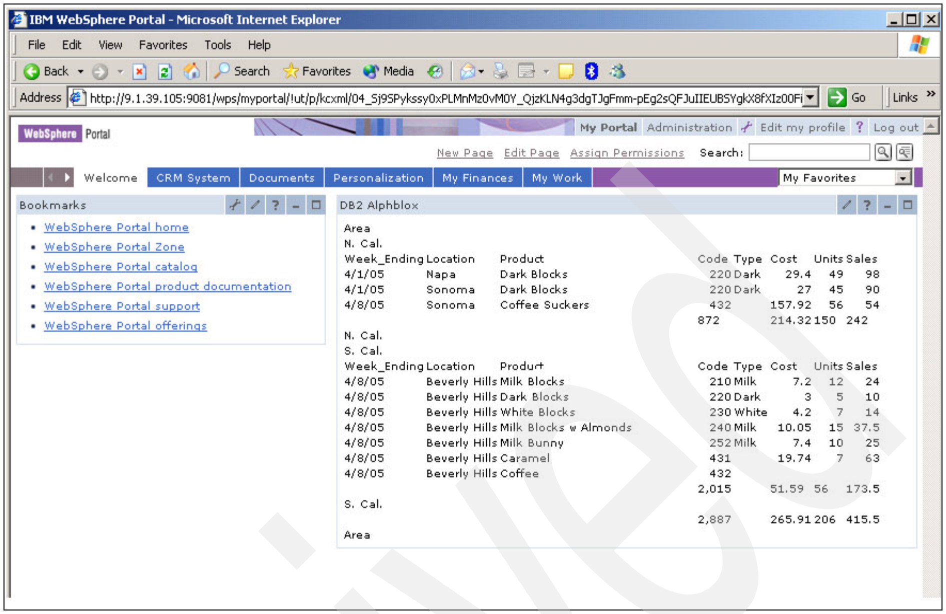


Figure 7-29 Alphablox portlet with CRM portlet

The Alphablox Portlet is only available on Fixpak 1 of DB2 Alphablox 8.2. To deploy this portlet in the case study environment, the following steps were used:

- Log on to the portal server as administrator.
- Click Administration and choose Web Modules to install the AlphabloxSamplePortlets.war file.
- Click Portal Settings and choose import XML option to import the sampleportlet.xml file.

The AlphabloxSamplePortlets.war file can be found on the Alphablox support page, as part of Fixpak 1.

Alphablox was enabled in the Portal through the installation of its portlet. It was installed by going to the Administration section as depicted in Figure 7-30.

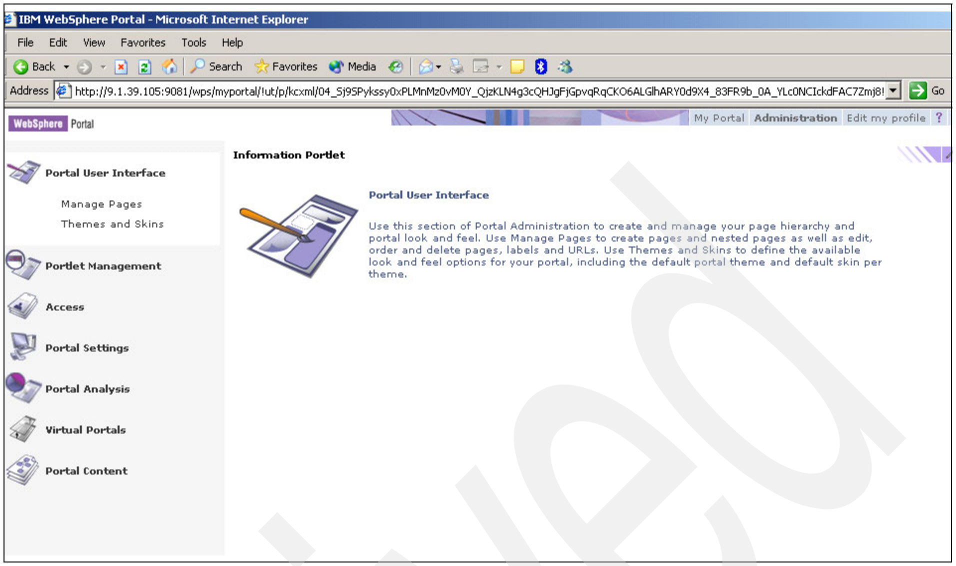


Figure 7-30 WebSphere Portal Administration section

Portlet management is the menu that must be selected, and then Add Web Module option. A screen will pop-up on the right side. Click Install, and select the WAR file provided with DB2 Alphablox Fixpak 1. This window is depicted in Figure 7-31.

Figure 7-31 Portlet Management

This first portlet displays the states of all processes within a specific time interval. The measure used is Process Instances, which shows how many instances of a particular process have completed, failed, or are still running. A PresentBlox was used that shows both the grid and chart of the same data. The number of failed processes is shown in red, to highlight them. The user can customize the appearance of reports interactively using the context menu of each report. The report is shown in Figure 7-32.

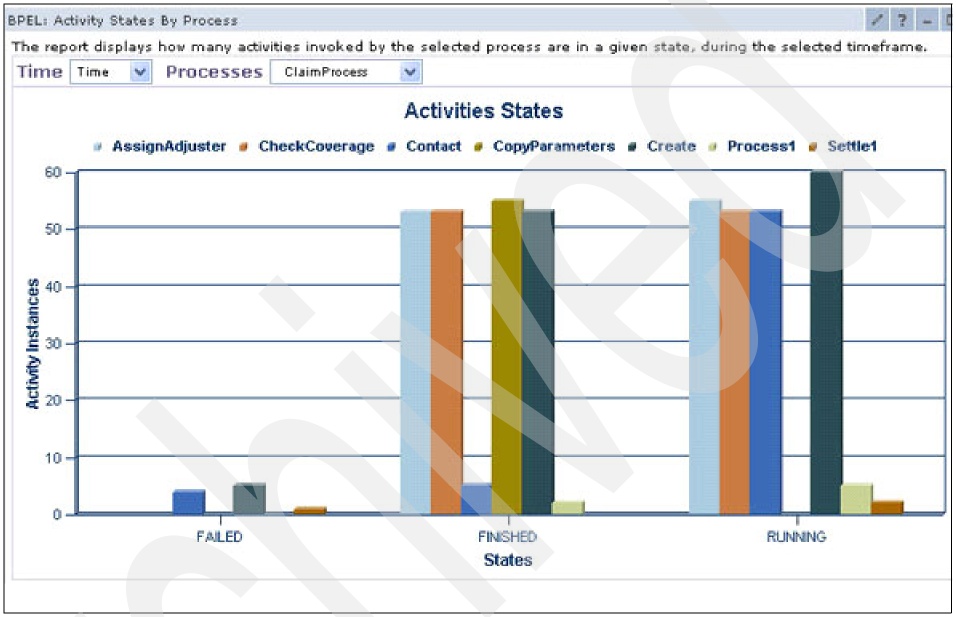


Figure 7-32 Process States Portlet

As shown in Example 7-6, the portlet JSP defines a Data Blox that connects to the cube, retrieves the appropriate data, and then supplies it to the Present Blox, rendering it using both a Grid and a Chart Blox.

Example 7-6 Example of data and present Blox for displaying process states

```
<blox:data

  bloxName="<%= dataBloxName %>" visible="false" dataSourceName="AuditCube"
  query="SELECT DISTINCT({[AuditCube].[Process States].children})ON AXIS(0),
             DISTINCT({[AuditCube].[Processes].children})ON AXIS(1)
             FROM [AuditCube] where [Process Instances]"
     ...
</blox:data>

<blox:present visible="false" bloxName="<%=bloxName%>"
   id="ProcessStatesBlox">
```

```
  <blox:grid><blox:cellFormatforeground="red"  scope="{Process
     States:FAILED}"/></blox:grid>
  <blox:chart title="Process States" chartCurrentDimensions="[AuditCube].
     [Processes]" ... />
  <blox:data bloxRef="<%=dataBloxName%>" />
     ...
</blox:present>
```

It is often useful to show the states of all the activities associated with a particular process type. You can use interportlet communication to link the process states reports with the activity states report. We added a hyperlink to each process in the process states report in order to link to the matching activity states report. The activity state report shows how many instances of the associated process activities are in a given state. Figure 7-33 shows the activities states for the Claim Process.

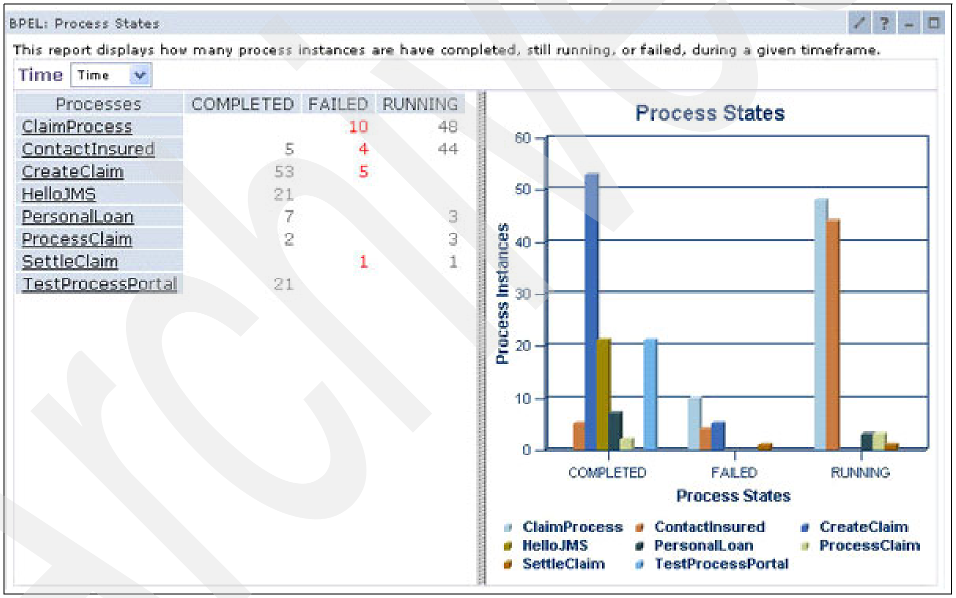


Figure 7-33 Activity states for the Claim Process

A hyperlink to each process is added by inserting the following code snippet into the present Blox for the Process States portlet as shown in Example 7-7.

Example 7-7 Insert a hyperlink in a present Blox

```
<blox:present visible="false" bloxName="<%=bloxName%>"
   id="ProcessStatesBlox">
     ...
  <bloxportlet:actionLinkDefinition action="setProcessName">
```

```
      <bloxportlet:parameter name="name"/>
  </bloxportlet:actionLinkDefinition>

<%
 PortletLink link  = ProcessStatesBlox.getPortletLink("setProcessName");
 BloxModel model   = ProcessStatesBlox.getGridBlox().getBloxModel();
 Controller controller   = model.getController();
 GridEventHandler eventHandler =  new GridEventHandler(link);
 controller.addEventHandler(eventHandler);
%>

</blox:present>
```

On the receiving end, the Activity States portlet needs to implement the standard actionPerformed() and messageReceived() methods specified in the IBM Portlet API to handle the interportlet communication. Typically, the name of the process can be set at the session scope (or whatever scope fits your application), and the activities states JSP page will customize the report to show the activities of the selected process.

For further information about how to set up this test, access the IBM Developerworks Web site for the article “Use DB2 Alphablox to generate reports for your business processes”, by using the following URL:

http://www-128.ibm.com/developerworks/ibm/library/i-odoebp12

7.2.4 Enterprise Application Integration

EAI is used in our test scenario. The configuration is depicted in Figure 7-34.

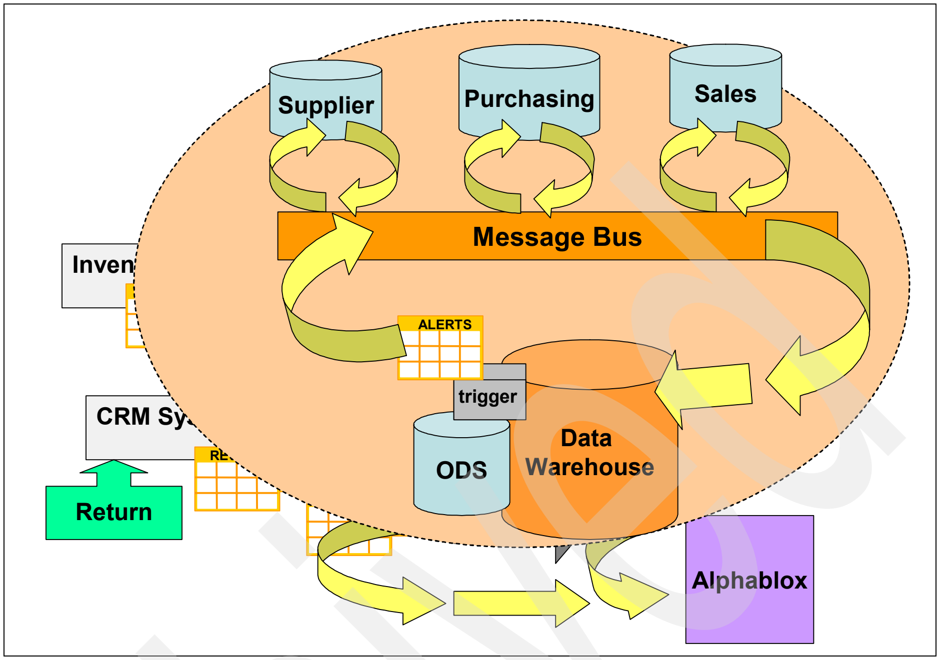


Figure 7-34 EAI in test scenario

The test cases are comprised of two WebSphere message flows. One is for Test Scenario A, whose main flow is depicted in Figure 7-35.

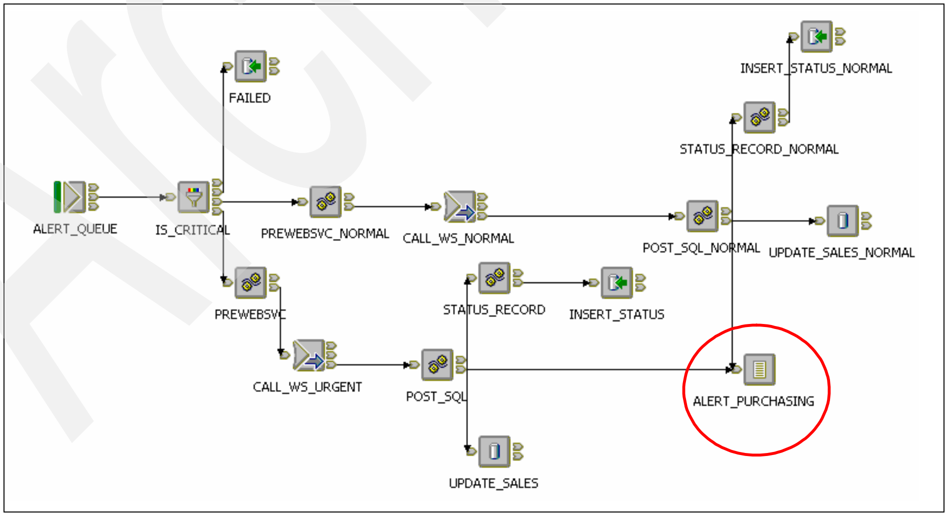


Figure 7-35 Main Flow

The second is a sub-flow, responsible for propagating the information to the Purchasing system, and it is depicted in Figure 7-36.

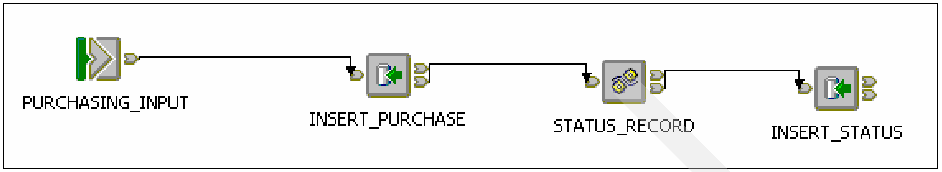


Figure 7-36 Purchasing Sub-Flow

In Figure 7-35 there is the main flow with several nodes. The highlighted node, called ALERT_PURCHASING, is responsible for invoking the Purchasing sub-flow. To enable a better understanding of the flows, we describe each node, and its functionality and type, in Table 7-1.

Table 7-1 Nodes description

Node Name	Node Type	Description
ALERT_QUEUE	MQInput	Reads the queue that receives publications from WebSphere II.
IS_CRITICAL	Filter	Check the amount of returns with a customized number. If the number of returns is greater, then it routes the flow to the critical path, otherwise it just routes it to the regular high returns message flow.
PREWEBSVC	Compute	Adds a critical flag on the message and creates the SOAP envelope that is used to invoke the Supplier Web service.
CALL_WS_URGENT	HTTPRequest	Requests the Web service providing the SOAP envelope with the information about the product ID, number of returns and description.
POST_SQL	Compute	Parses the Web services output.
UPDATE_SALES	Database	Updates the sales database, putting any sales containing the defective product ID on hold.

Node Name	Node Type	Description
STATUS_RECORD	Compute	Adds an order field in the message to insert data on the ALERT_STATUS table, which is used to generate reports on demand.
INSERT_STATUS	DataInsert	Inserts data on the ALERT_STATUS.
ALERT_PRUCHASING	Sub-flow	Invokes the Purchasing Sub-flow.
PREWEBSVC_NORMAL	Compute	Creates the SOAP envelope that is used to invoke the Supplier Web service.
CALL_WS_NORMAL	HTTPRequest	Requests the Web service providing the SOAP envelope with the information about the product ID, number of returns and description.
POST_SQL_NORMAL	Compute	Parses the Web services output.
UPDATE_SALES_NORMAL	Database	Updates the sales database, putting all sales containing the defective product ID on hold.
STATUS_RECORD_NORMAL	Compute	Adds an order field in the message to insert data on the ALERT_STATUS table, which is used to generate reports on demand.
INSERT_STATUS_NORMAL	DataInsert	Inserts data on the ALERT_STATUS.
PURCHASING_INPUT	Input	Sub-flow input.
INSERT_PURCHASE	DataInsert	Inserts data in the Purchasing system.
STATUS_RECORD	Compute	Adds an order field in the message to insert data on the ALERT_STATUS table, which is used to generate reports on demand.
INSERT_STATUS	DataInsert	Inserts data on the ALERT_STATUS.

Nodes can represent a logical situation. At times, one node is not sufficient and a group of nodes must be created. Then, together they represent the entire process. In the test case, we created a logical flow of the message flow created, as shown in Figure 7-37.

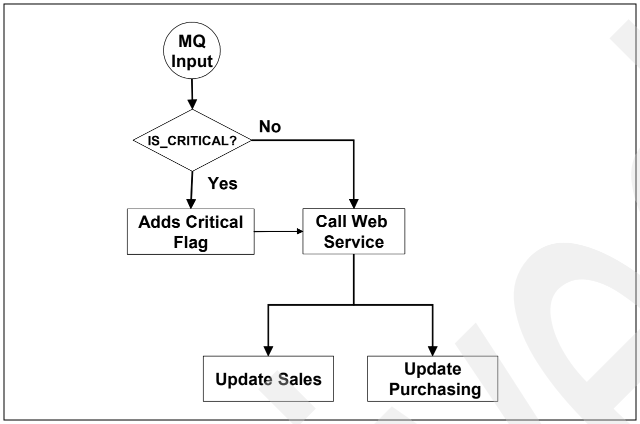


Figure 7-37 Logical flow

Based on the logical flow and the message flow described, we now describe how the WebSphere ESB test was planned, created and executed.

In Test Scenario A, WebSphere II Event Publisher was configured and set on the ODS. The publishing should replicate all new records that are inserted in the table named ALERT that receive a new entry every time a product is returned and the total is more than the threshold value. This insertion is done through the use of triggers on every insertion executed on the returns table. It counts the number of returns due to defect, on the ODS, and if that number is above the threshold value, a record is inserted in the ALERT table. Then the WebSphere II Event Publisher publishes this new entry to a queue that is monitored by the WebSphere Message flow. For more details about WebSphere Event Publishing and how the triggers were configured, see “Event Publishing” on page 305.

The information that WebSphere II Event Publisher sends to a queue is formatted as an XML message. An entry publication was captured and can be seen in Example 7-8.

Example 7-8 XML generated by WebSphere II Event Publisher

```
<?xml version="1.0" encoding="UTF-8" ?>
<msg xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="mqcap.xsd" version="1.0.0" dbName="DW">
   <rowOp authID="DB2INST1" cmitLSN="0000:0000:0000:a711:9407"
cmitTime="2005-11-10T10:51:09.000005">
      <insertRow subName="ALERT0001" srcOwner="DWH" srcName="ALERT">
```

```
         <col name="PD_ID" isKey="1">
            <integer>33141</integer>
         </col>
         <col name="DESCRIPTION">
            <varchar>THIS PRODUCT IS WITH A HIGH NUMBER OF RETURNS, SHOULD BE 
INVESTIGATED</varchar>
         </col>
         <col name="NUMBER_RETURNS">
            <integer>5</integer>
         </col>
         <col name="TIME">
            <timestamp xsi:nil="2005-11-10-10.44.07.761671"/>
         </col>
      </insertRow>
   </rowOp>
</msg>
```

To create the message file definition inside WebSphere, and to be able to recognizes the message that we are gathering on the flow, we had to generate a XML Schema file, a XML Schema file that is a valid structure and allows a message definition file creation. There are two XML Schema files generated. The first is shown in Example 7-9.

Example 7-9 mq_schema.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" 
elementFormDefault="qualified" 
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <xs:import namespace="http://www.w3.org/2001/XMLSchema-instance" 
schemaLocation="xsi.xsd"/>
  <xs:element name="msg">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="rowOp"/>
      </xs:sequence>
      <xs:attribute name="dbName" use="required" type="xs:NCName"/>
      <xs:attribute name="version" use="required" type="xs:NMTOKEN"/>
      <xs:attribute ref="xsi:noNamespaceSchemaLocation" use="required"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="rowOp">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="insertRow"/>
      </xs:sequence>
      <xs:attribute name="authID" use="required" type="xs:NCName"/>
```

```
      <xs:attribute name="cmitLSN" use="required" type="xs:NMTOKEN"/>
      <xs:attribute name="cmitTime" use="required" type="xs:NMTOKEN"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="insertRow">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="col"/>
      </xs:sequence>
      <xs:attribute name="srcName" use="required" type="xs:NCName"/>
      <xs:attribute name="srcOwner" use="required" type="xs:NCName"/>
      <xs:attribute name="subName" use="required" type="xs:NCName"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="col">
    <xs:complexType>
      <xs:choice>
        <xs:element ref="integer"/>
        <xs:element ref="timestamp"/>
        <xs:element ref="varchar"/>
      </xs:choice>
      <xs:attribute name="isKey" type="xs:integer"/>
      <xs:attribute name="name" use="required" type="xs:NCName"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="integer" type="xs:integer"/>
  <xs:element name="timestamp" type="xs:NMTOKEN"/>
  <xs:element name="varchar" type="xs:string"/>
</xs:schema>
```

The second XML Schema file is shown in Example 7-10.

Example 7-10   xsi.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified"
targetNamespace="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <xs:import schemaLocation="mq_schema.xsd"/>
  <xs:attribute name="noNamespaceSchemaLocation" type="xs:NCName"/>
</xs:schema>
```

As previously mentioned, the Schema Files can be transformed to a message file definition. Based on the schema files generated, we created two message definition files shown. The first is depicted in Figure 7-38.

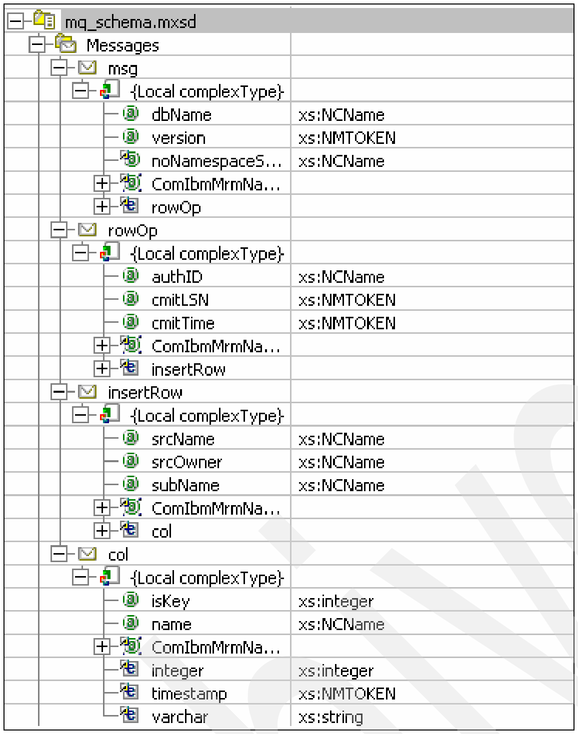


Figure 7-38 mq_schema.mxsd

The second message definition file is depicted in Figure 7-39.

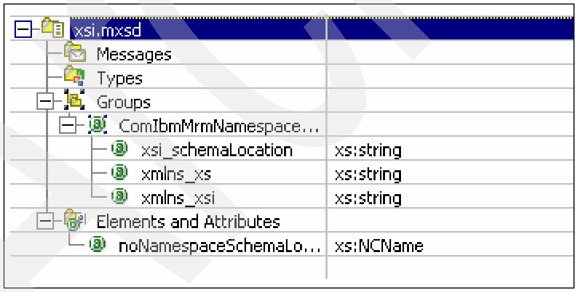


Figure 7-39 xsi.mxsd

These message definition files are used to parse the message that the flows receive in the queue. Another two message definitions were created to assist in the flow development, they are the alert message and the alert_control message.

The alert message is based on the same structure as the alert table. The only difference is an additional boolean field that determine if it is a critical or a non-critical situation. This message definition has the following fields:

- product_id: contains the product ID number.
- number_returns: number of returns due to technical problems.
- description: description about the returns.
- critical: if the number of returns were above a customized value, this flag is set to true.
- alert_id: the alert ID number, used to insert tracking data on the alert_control table and consequently generate the tracking reports.

The alert_control message is used to store tracking data about how the flow is processed. This table is then used to generate reports about how the high number of returns were handled by the flow. This message definition has the following fields:

- alert_id: the alert ID number, used to insert tracking data on the alert_control table and consequently generate the tracking reports.
- control_alert_order: the order number that the node that is storing tracking information was executed. For example, the first tracking node will add it as order number 1, the second one as order number 2, and so on.
- control_alert_description: brief description of how it was executed and the return code of the execution.

The alert message definition file and Alert_Control message definition file is shown in Figure 7-40.

alert.mxsd	
Messages	
ALERT	ALERT_type
product_id	xsd:int
number_returns	xsd:int
description	xsd:string
critical	xsd:boolean
alert_id	xsd:int
ALERT_CONTROL	ALERT_CONTROL
ALERT_ID	xsd:string
CONTROL_ALERT_ORDER	xsd:string
CONTROL_ALERT_DESCRIPTION	xsd:string

Figure 7-40 alert and alert_control messages

In Test Scenario B, the flow is nearly the same as in Scenario A. The only change was the first node, which is an HTTP Input node. And at the message end, an HTTP Reply. The HTTP Input and HTTP Reply node properties are depicted in Figure 7-41. The flow was changed from its originally form to allow DB2 to connect directly to the ESB.

HTTP Input

Here we use the HTTP Input node to receive Web service requests for processing by a message flow. Using the HTTP Input node with the HTTP Reply and HTTP Request nodes, the broker can act as an intermediary for Web services, and Web service requests can be transformed and routed in the same way as other message formats supported by WebSphere Message Broker. Web service requests can be received either in standard HTTP (1.0 or 1.1) format, or in HTTP over SSL (HTTPS) format. You can set the *Use HTTPS* property to choose whether to handle HTTP or HTTPS requests.

If you include an HTTP Input node in a message flow, you must either include an HTTP Reply node in the same flow, or pass the message to another flow that includes an HTTP Reply node (for example, through an MQOutput node to a second flow that starts with an MQInput node). In the latter case, the request from, and reply to, the client are coordinated by the request identifier stored in the LocalEnvironment.

The HTTP Input node handles messages in the following message domains:

- MRM
- XML
- XMLNS
- XMLNSC
- JMS
- JMSStream
- IDOC
- MIME
- BLOB

When the HTTP Input node receives a message from a Web service client, it invokes the appropriate parsers to interpret the headers and the body of the message, and to create the message tree that is used internally by the message flow. The node creates a unique identifier for the input message and stores it as a binary array of 24 bytes in the LocalEnvironment tree at *LocalEnvironment.Destination.HTTP.RequestIdentifer*. This value is used by the HTTP Reply node and must not be modified in any way.

HTTP messages are always non-persistent, and have no associated order. They are also non-transactional. However, if the message flow interacts with a database or another external resource such as a WebSphere MQ queue, these interactions are performed transactionally. The HTTP Input node provides commit or rollback depending on how the message flow has ended, and how it is configured for error handling (how failure terminals are connected, for example).

If the message flow is rolled back by this node, a fault message is generated and returned to the client. The format of the fault is defined by the Fault Format property.

The HTTP input node properties used in Test Case Scenario B, are depicted in Figure 7-41.

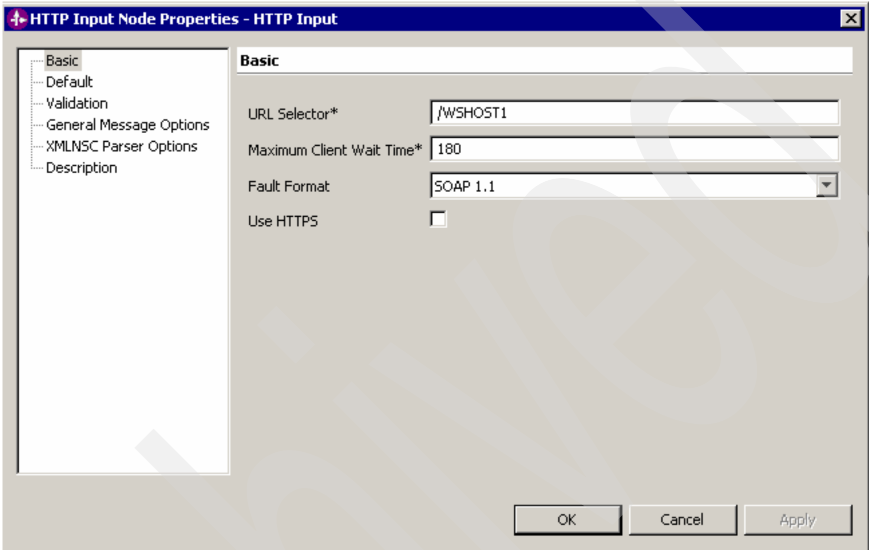


Figure 7-41 HTTP Input Properties

HTTP Reply

Use the HTTP Reply node to return a response from the message flow to the Web service client. This node generates the response to the Web service client from which the input message was received by the HTTP Input node, and waits for confirmation that it has been sent.

If you include an HTTP Reply node in a message flow, you must either include an HTTP Input node in the same flow, or the message must be received from another flow that started with an HTTP Input node. The response is associated with the reply by a request identifier that is stored in LocalEnvironment by the HTTP Input node.

This node constructs a reply message for the Web service client from the entire input message tree, and returns it to the requestor. The HTTP Reply node properties used in Test Scenario B are depicted in Figure 7-42.

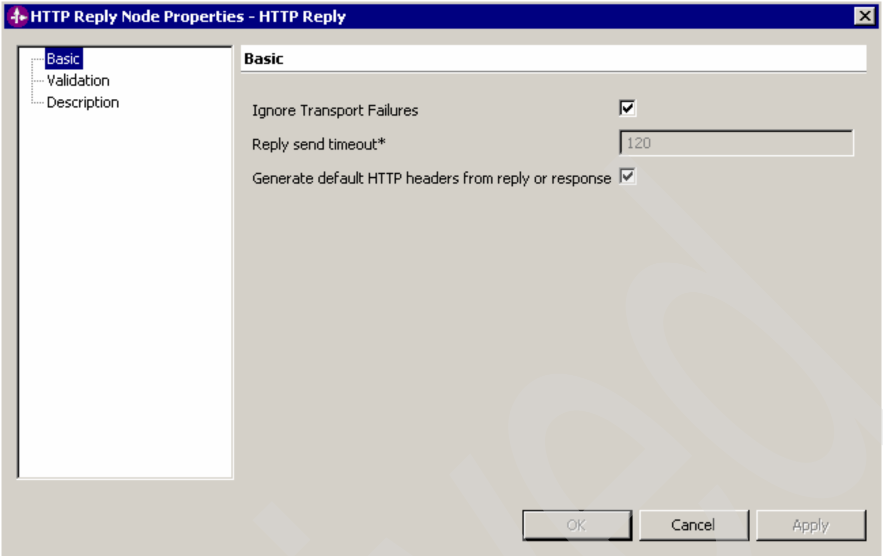


Figure 7-42 HTTP Reply properties

The sales and purchasing system are being fed with information about the high number of returns of a specific product, through insertions in tables belonging to each system. Although this is a good solution for some systems, it is not good enough to integrate all systems in a real-time enterprise. Inside WebSphere there are several ways to access and retrieve data. However, due to time constraints and the scope of this book, we do not describe all of them. Rather, we have chosen one that is very popular - Web services. In our test case, the supplier system is hosted outside the corporation and Web services is the interface to integrate with it. Now we can take a look on how we implemented this in the flow.

Inside the message flow we invoke the Web services using an HTTP Request node. To do that, we have to encapsulate the information in a SOAP envelope which is used to pass information to the Web services. This is depicted in Figure 7-43.

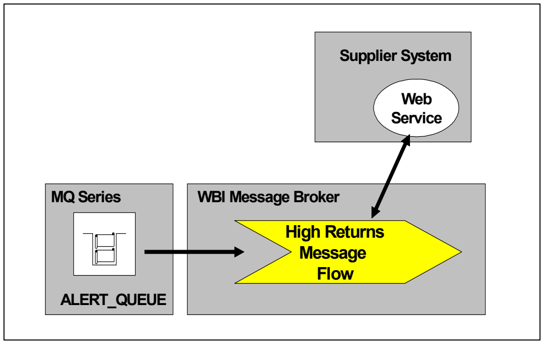


Figure 7-43 Supplier Web service

To create the test case and test it, we developed a simple Web service called *Supplier_returns* to simulate the supplier communication interface. After that, we deployed this application in a WebSphere Application Server and left it available to the WebSphere flow. In Example 7-11, Example 7-12 and Example 7-13 are the supplier Web services generated WSDL (Web service description language).

Example 7-11 Supplier_returns.wsdl

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="Supplier_returns"
    targetNamespace="http://itso.ibm.com/Supplier_returns/"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:tns="http://itso.ibm.com/Supplier_returns/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <message name="insertReturnRequest">
        <part name="productid" type="xsd:int"/>
        <part name="returns" type="xsd:int"/>
        <part name="critical" type="xsd:boolean"/>
        <part name="description" type="xsd:string"/>
        <part name="alert_id" type="xsd:int"/>
    </message>
    <message name="insertReturnResponse">
        <part name="result" type="xsd:int"/>
    </message>
    <portType name="Supplier_returns">
        <operation name="insertReturn" parameterOrder="productid returns
critical description alert_id">
            <input message="tns:insertReturnRequest"
name="insertReturnRequest"/>
            <output message="tns:insertReturnResponse"
name="insertReturnResponse"/>
        </operation>
    </portType>
```

```
</definitions>
```

In Example 7-12 we describe the SOAP Web service description language (WSDL).

Example 7-12 Supplier_returnsSOAPService.wsdl

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="Supplier_returnsService"
    targetNamespace="http://itso.ibm.com/Supplier_returnsSOAPService/"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:binding1="http://itso.ibm.com/Supplier_returnsSOAPBinding/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://itso.ibm.com/Supplier_returnsSOAPService/">
    <import location="Supplier_returnsSOAPBinding.wsdl"
namespace="http://itso.ibm.com/Supplier_returnsSOAPBinding/"/>
    <service name="Supplier_returnsService">
        <port binding="binding1:Supplier_returnsSOAPBinding"
name="Supplier_returnsSOAPPort">
            <soap:address
location="http://supplier.itso.ibm.com/WebServicesWeb/servlet/rpcrouter"/>
        </port>
    </service>
</definitions>
```

In Example 7-13 we describe the SOAP Binding Web service description language (WSDL).

Example 7-13 Supplier_returnsSOAPBinding.wsdl

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="Supplier_returnsJavaBinding"
    targetNamespace="http://itso.ibm.com/Supplier_returnsJavaBinding/"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:format="http://schemas.xmlsoap.org/wsdl/formatbinding/"
    xmlns:interface1="http://itso.ibm.com/Supplier_returns/"
    xmlns:java="http://schemas.xmlsoap.org/wsdl/java/"
    xmlns:tns="http://itso.ibm.com/Supplier_returnsJavaBinding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <import location="Supplier_returns.wsdl"
namespace="http://itso.ibm.com/Supplier_returns/"/>
    <binding name="Supplier_returnsJavaBinding"
type="interface1:Supplier_returns">
        <java:binding/>
        <format:typeMapping encoding="Java" style="Java">
            <format:typeMap formatType="int" typeName="xsd:int"/>
```

```
            <format:typeMap formatType="java.lang.String"
typeName="xsd:string"/>
            <format:typeMap formatType="boolean" typeName="xsd:boolean"/>
        </format:typeMapping>
        <operation name="insertReturn">
            <java:operation methodName="insertReturn"
                parameterOrder="productid returns critical description
alert_id" returnPart="result"/>
            <input name="insertReturnRequest"/>
            <output name="insertReturnResponse"/>
        </operation>
    </binding>
</definitions>
```

Using a tunneling tool, we captured the SOAP envelope which would be necessary to call the Web service from the WebSphere flow. The HTTP Request node is configured to access the server *supplier.itso.ibm.com*, which is the Supplier server in our environment. In the flow, CALL_WS_URGENT and CALL_WS_NORMAL access the same Web service and is shown in Figure 7-44.

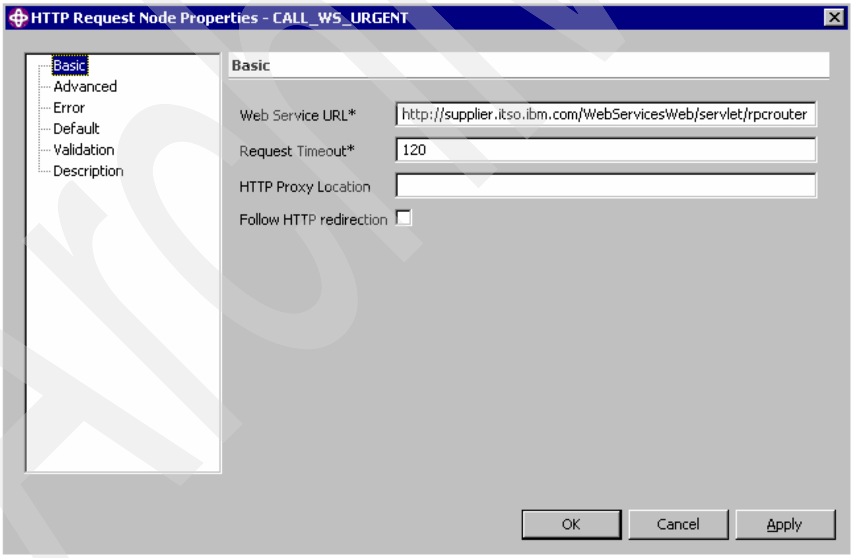


Figure 7-44 CALL_WS_URGENT node details

At this point we have discussed Test Scenario A, with WebSphere Message Broker V5. Next we move to Test Scenario B, that uses V6 of WebSphere

Message Broker. It has the Advanced Enterprise Service Bus capabilities, as well as some new features such as:

- A Java compute node which provides node capabilities for Java programmers.
- Timer Control node.
- MQGET node which supports SOAP/JMS(MQ).
- JMS Input/Output native JMS interoperability.

There are also some new mechanisms, such as a new WSDL importer wizard that accepts a variety of WSDL styles.

The Enterprise Service Bus (ESB) is a significant capability and advantage of WebSphere. It has an architectural pattern that supports virtualization and management of service interactions. It also provides connectivity between requesters and providers (based on SOA) and can be implemented using a variety of middleware technologies and programming models.

Using the new version, we are able to add new services even faster by plugging the service into the ESB, which improves changes and maintenance time and costs, minimizing the impact to existing services.

Test Scenario B relies on a different approach for integrating the ODS with other applications and services inside and outside the enterprise. Using the built-in features of DB2 UDB, it enables the access to one WebSphere ESB message flow through the use of Web services. That is accomplished by an HTTP Input node. This scenario is depicted in Figure 7-45.

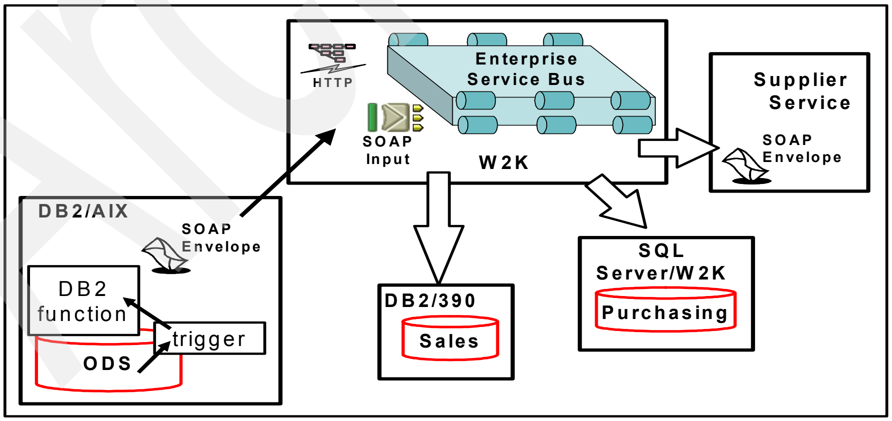


Figure 7-45 DB2 and ESB flow

We created triggers in the ODS which are responsible for identifying when there are a high number of returns. When that occurs the DB2 function we created (call_Svce_High_Ret) is executed using DB2 Web services. The triggers are described in "Triggers" on page 310.

7.2.5 Test with combined EII and ETL

We start by highlighting the points of integration between WebSphere DataStage and WebSphere Data Federation, and then we illustrate the integration using a simple business scenario in which customer databases from two companies are merged.

This enables users to seamlessly get reports displaying data that is coming concurrently from the data warehouse and from the operational data sources. The data from the operational sources is being cleansed and transformed in real-time using the same cleansing and transformations type of jobs that are invoked for building the data warehouse. These jobs are exposed in real-time as services using WebSphere Real Time Integration (RTI). Regarding data federation, the database administrator defines nicknames for the various data sources, including Web services, XML files, DB2 databases, and an IBM IMS database. This makes all data sources appear as standard database tables.

Transformation is sometimes only visible on the DataStage flows but it may also occur on the Replication which is responsible to cast data types or even transform and integrate data from distinct character set sources such as ASCII and EBCDIC, and vice-versa.

We tested two scenarios where the use of WebSphere Information Integration tools yielded significant benefits:

- Federation was used as to seamlessly integrate disparate data sources on the DataStage flows.
- Federation enabled access to shared transformation and data quality services from WebSphere DataStage, as a real-time data source, by using Web services wrappers.

The benefits of using Federation as a source to WebSphere DataStage includes:

- Seamless access to more information sources, including the mainframe and unstructured content.
- A single query and single retrieval step across heterogeneous data sources.
- Caching of commonly accessed information to improve response times and reduce the impact on source systems.
- No need to create intermediate file structures or to merge data before making it available for profiling, or standardization and matching.

- Fewer steps and more re-use results in less maintenance, less time, and increased productivity.
- Single vendor solution, and packaged on the same family of products of WebSphere Information Integration.

Using Replication to access transformation and data quality services from WebSphere DataStage as a real-time source empowers businesses with these capabilities:

- Embedding of complex transformation logic in federated queries.
- Applying data quality rules to federated data.
- Re-use of existing batch oriented transformation rules.
- Combinations with other real-time sources.
- Publishing of logic through an SQL interface.

In this section, we use parts of our environment to show an integrated test of a WebSphere DataStage ETL process and the WebSphere Information Integrator federation approach.

Since the ValueTrend company has gone through recent acquisitions, the inquiry about recent orders must be performed against the operational data of ValueTrend and the two companies acquired, WorldCo and GlobalCo.

In Figure 7-46 we show a overview of the systems used in the case study.

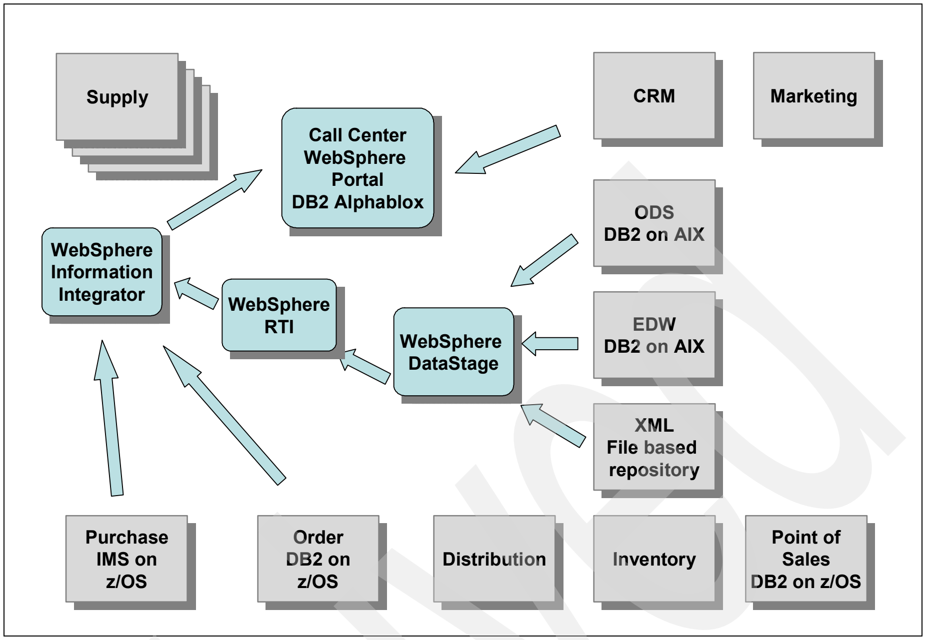


Figure 7-46 The environment for an EII and ETL combined test.

The business scenario for the demo revolves around a customer making an inquiry about very recent orders. Two queries will be issued by the clerk at the company's call center to check on this customer:

1. The first query verifies the orders submitted today by the customer. It runs against the operational data stores because the requested information has not yet been loaded in the data warehouse.
2. The second query compares today's aggregated sales data for the customer with his purchase history. This query needs to collect data from both the operational data store(s) and the data warehouse.

Query 1

The first query is:

```
Which orders did the customer submit today?
```

The operational data is stored in relational databases. However, as is very often the case, this operational data is off-limits for general scan-type queries. The way this operational data is made available for general consumption by users via a message-queuing infrastructure.

The output of the message queues consists of XML messages. So, for our purpose, the operational data are the XML files. Each operational transaction results in a message being put on the queue and then being consumed by applications as individual XML files. A sample of a XML file is shown in Example 7-14.

Alternatively, it is also a common practice to have a changed data capture, such as data replication technology, used against the operational data, and the changed data captured can then be published as XML to a message queue. For more information, refer to “Data Replication” on page 298 and “Event Publishing” on page 305.

Example 7-14 Sample file of a XML data record

```
<?xml version="1.0" encoding="UTF-8" ?>
<GLOBAL_SALES_TRAN>
<DETAIL>
  <SHIP_TO_NUMBER>10117</SHIP_TO_NUMBER>
  <SALES_ORDER_NUMBER>1</SALES_ORDER_NUMBER>
  <LINE_NUMBER>9082929 16</LINE_NUMBER>
  <CASES_SHIPPED>20</CASES_SHIPPED>
  <GROSS_SALES>347.68</GROSS_SALES>
 </DETAIL>
 </GLOBAL_SALES_TRAN>
```

The data in the XML files needs to be transformed, in the same way as it is transformed before being loaded in the data warehouse. For example, converting the customer operational codes into real customer names for billing and shipping.

This transformation is published by WebSphere SOA Edition as a Web service for real-time transformation of the operational data.

In Figure 7-47 we show the WebSphere DataStage transformation that we used in our test scenario. We show WebSphere DataStage implemented lookup to the dimension table in the data warehouse, which is the service *GetKeysRealTime* that is published by WebSphere Data Integration SOA Edition. The GetKeysRealTime job returns the surrogate key values, ShipToKey and BillToKey, associated to the input ShipToNumber.

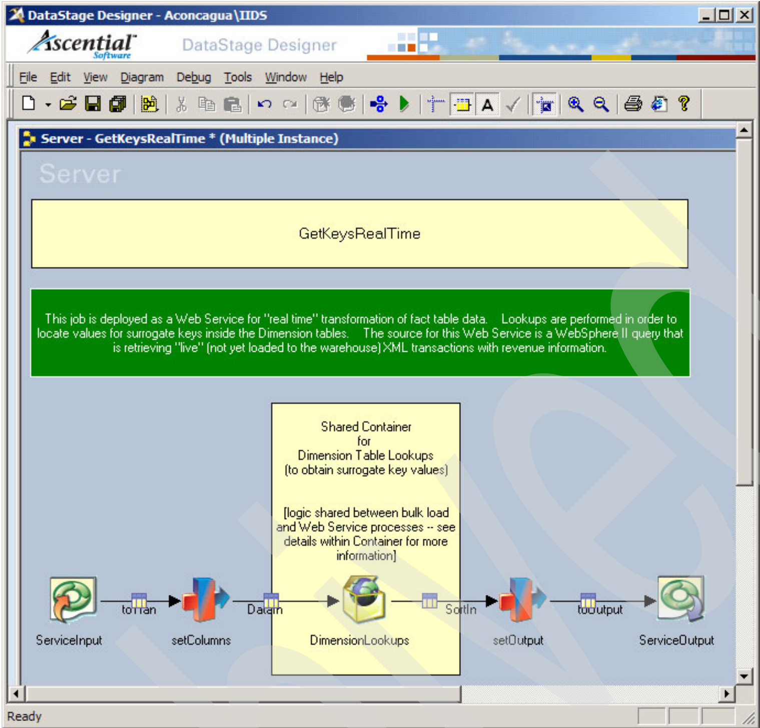


Figure 7-47 WebSphere DataStage data transformation

In Figure 7-48 we show how the customer name and address can then be retrieved from the data warehouse *BillTo_Dimension* table using the surrogate keys delivered as output of the service.

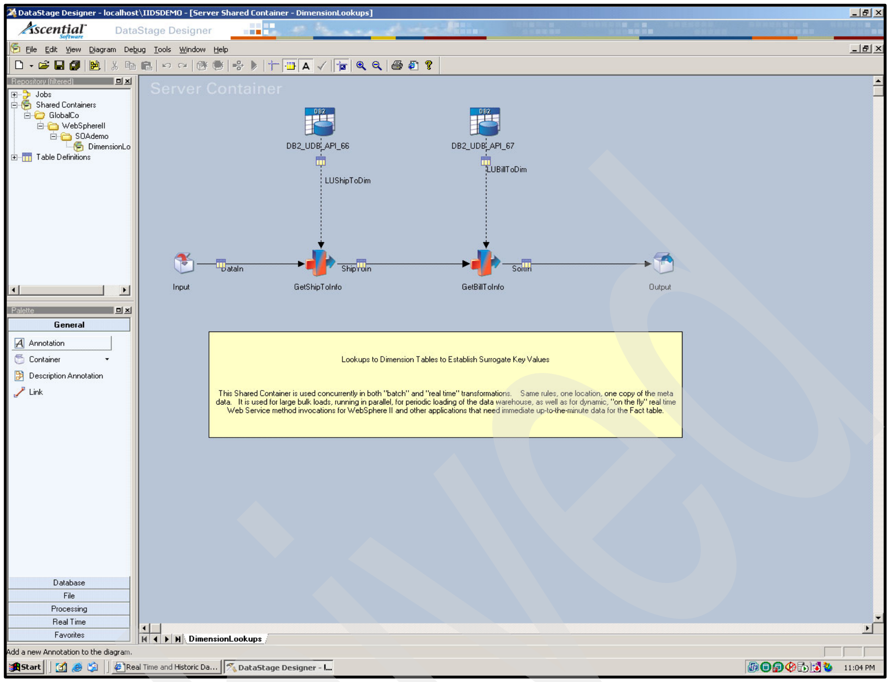


Figure 7-48 WebSphere DataStage dimension lookups

In the next section we describe how WebSphere Information Integrator can greatly help in combining all these operations in a single query.

Query 2

The second query is:

```
How do today's orders made by this customer compare with his past orders?
```

The result of the first query provided a detailed report about the transactions performed today by the customer. This information was extracted from the operational data stores, since it has not yet been consolidated in the data warehouse. However, we used in real-time the transformations that would have been invoked at the time the data warehouse would be updated with this recent data.

Now, we are interested in having a summary of today's customer activity, and compare it with the past activity. To do this we use a query very similar to the first one, but we aggregate the results and compare them (via a UNION statement) with the past activity for the customer. This is information that resides in the data warehouse.

The result of the query should help the call center operator decide whether some extra care should be given to this customer, if for example he has generated a substantial amount of business with the company in the past.

WebSphere Information Integrator lets us combine all this information into a single query, as shown in Example 7-15. Without it, separate reports would need to be generated, and data related to the same customer would not be directly apparent because different codes are used in the operational data and in the data warehouse to identify the customer.

Example 7-15 Single SQL that combines extract from different data sources

```
SELECT parameters_return_billto_key as BILL_TO_KEY,
       billto_company_name, parameters_return_shipto_key as SHIP_TO_KEY,
       int(ltrim(rtrim(translate(cases_shipped, ' ', x'0a')))) as
CASES_SHIPPED,
      decimal(ltrim(rtrim(translate(gross_sales, ' ', x'0a')))) as GROSS_SALES
FROM   GETKEYSSOAP_GETKEYSREALTIME_NN,
       GLOBAL_SALES_TRAN_NN,
       BILLTO_DIMENSION
WHERE  getkeysrealtime_ship_to_number = '13546'
   and ltrim(rtrim(translate(ship_to_number, ' ', x'0a'))) =
getkeysrealtime_ship_to_number
   and parameters_return_billto_key = billto_key;
```

DB2 Alphablox reports

As described in 7.2.3 “The dashboard” on page 317, we deployed DB2 Alphablox in the WebSphere Portal Server. With these few Alphablox reports we try to illustrate an analytic environment that could be used to deliver both executives dashboard information as well enabling them to drill deeper into the sales statistics by store, brand, product, as examples.

These reports could, among other purposes, be used as part on the decision making process with our previously mentioned faulty hair dryer scenario. The decision to take the hair dryer off the market and stop production, and the final decision to offer customers the capability to return the hair dryers and receive a full refund are all decisions that need extensive analytics of sales information, for example.

In Figure 7-49 we show an example of a Trend Analysis report that can be used to see actual sales in comparison with the planned sales for a given period.

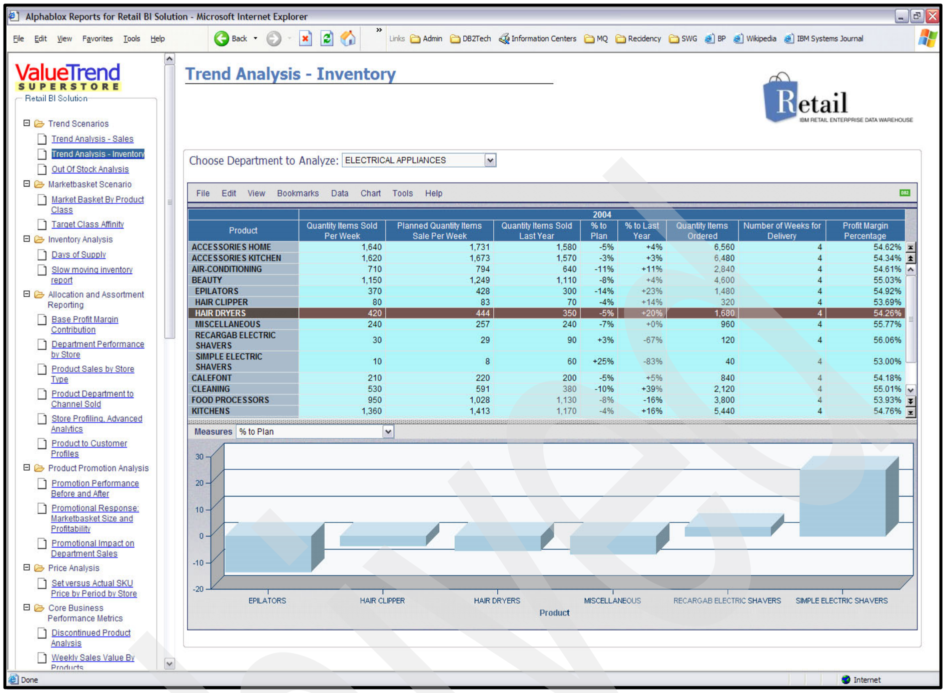


Figure 7-49 Sample report for Trend Analysis of Inventory

Another example of a DB2 Alphablox report that we evaluated in our retail scenario, could be similar to the report shown in Figure 7-50 where we breakdown the product sales by department and customer segments.

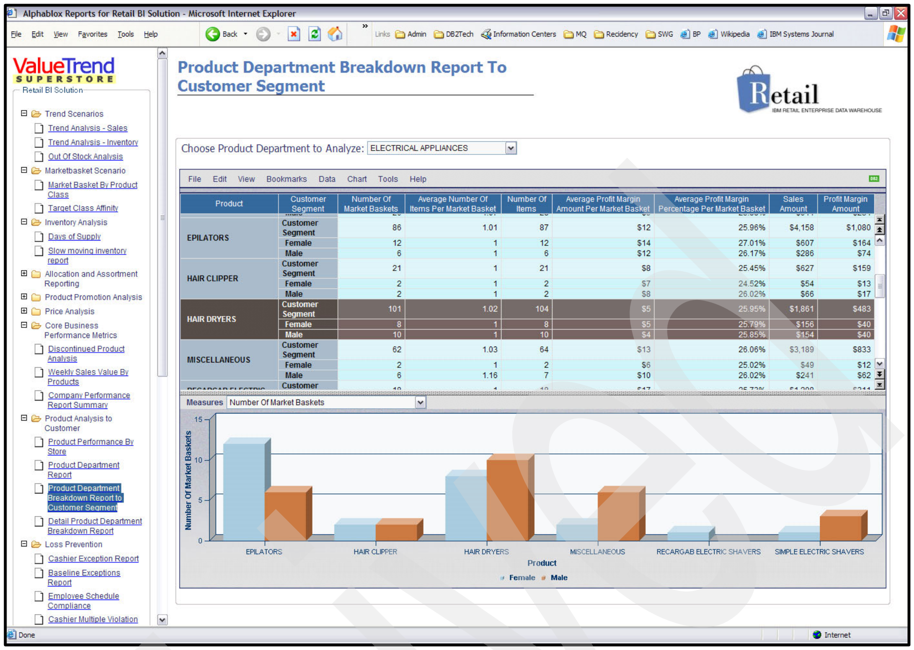


Figure 7-50 Product by Department and customer segment breakdown

Figure 7-51 depicts our last example DB2 Alphablox report. It is the sales statistics for the hair dryer across the store types.

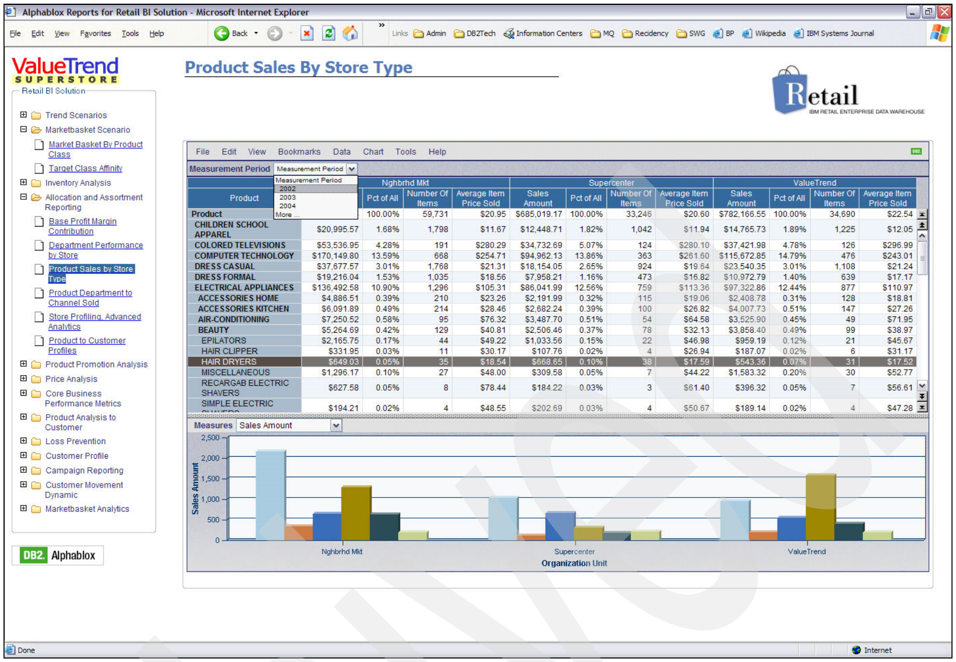


Figure 7-51   DB2 Alphablox of product sales by store type

These examples have all illustrated how we used the ODS and data warehouse in combination for both operational reporting and strategic analysis.

With the technologies used in the case study, it would be very possible to access information outside the ODS and the data warehouse. By using WebSphere II with the federation facilities, we can connect to any of the previously described data sources. By using the test integration technologies, combined with DB2 Alphablox, we can achieve powerful analytic capabilities by accessing the ODS and data warehouse together with the real-time access to a wide variety of information sources.

For these DB2 Alphablox reports, we used predefined reports and associated test data developed by IBM Retail Business Intelligence Solution. This solution uses a pre-existing set of products to enable retail businesses to build a cost-effective, resilient data infrastructure. It provides retailers with an information environment that can generate insight into all aspects of the core business. The sample reports in this demo cover some of the subject areas in the IBM Retail Data Warehouse model and also shows examples of advanced analytics for retailer-shaped data mining.

7.2.6 Closing the loop

We have, in this chapter, described the individual tests that enabled us to access and move information in a real-time enterprise environment. As part of the closed-loop test we used a simple scenario with trigger-based events being published to an entry in our *Alert* table. The alerts were then sent by WebSphere II Event Publishing to WebSphere Enterprise Service Bus, or in our alternative test, through a trigger-based event which calls the Web services features in DB2 that invoked the WebSphere Enterprise Service Bus. A combination of technologies have enabled us to integrate systems by moving information in real-time, or access it with near zero-latency, anywhere within or outside the retail organization.

These tests show how we can perform closed-loop processing, for example, with the supplier of the hair dryer. The supplier in the scenario is responsible to act on the information being provided, either through WebSphere or with the service oriented approach via the ESB. Our focus area has been about testing integration and then evaluating the latency profile for the data flow between environments, rather that focusing on the applications use of the data.

All of the tested technologies can, with proper setup and usage, deliver real-time, or near real-time, information. This leaves the IT organization with the task of choosing the appropriate technology to solve their integration challenges.

As we have shown with the test scenarios illustrated in this chapter, there are a number of approaches for implementing a real-time enterprise environment. It could be in the form of using traditionally developed data transfers, file transfer tools such as FTP, message queuing, data replication, event publishing, and ETL, using an information integration engine with real-time data federation, a message broker such as WebSphere, or last, but not least, the new generation of service oriented technologies such as the WebSphere ESB.

The capabilities of these individual technologies differ very much in implementation flexibility and effort. Therefore, an integration technology should always be validated for individual tasks and associated requirements.

Glossary

Access Control List (ACL). The list of principals that have explicit permission (to publish, to subscribe to, and to request persistent delivery of a publication message) against a topic in the topic tree. The ACLs define the implementation of topic-based security.

Additive Measure. Measure of a fact that can be added across all dimensions.

Aggregate. Pre-calculated and pre-stored summaries, kept in the data warehouse to improve query performance

Aggregation. An attribute level transformation that reduces the level of detail of available data. For example, having a Total Quantity by Category of Items rather than the individual quantity of each item in the category.

Analytic. An application or capability that performs some analysis on a set of data.

Application Programming Interface. An interface provided by a software product that enables programs to request services.

Associative entity. An entity created to resolve a many-to-many relationship into two one-to-many relationships.

Asynchronous Messaging. A method of communication between programs in which a program places a message on a message queue, then proceeds with its own processing without waiting for a reply to its message.

Attribute. A characteristic of an entity, such as a field in a dimension table.

BLOB. Binary Large Object, a block of bytes of data (for example, the body of a message) that has no discernible meaning, but is treated as one solid entity that cannot be interpreted.

Business subject area. A particular function or area within an enterprise whose processes and activities can be described by a defined set of data elements.

Candidate key. One of multiple possible keys for an entity.

Commit. An operation that applies all the changes made during the current unit of recovery or unit of work. After the operation is complete, a new unit of recovery or unit of work begins.

Compensation. The ability of DB2 to process SQL that is not supported by a data source on the data from that data source.

Composite Key. A key in a fact table that is the concatenation of the foreign keys in the dimension tables.

Computer. A device that accepts information (in the form of digitalized data) and manipulates it for some result based on a program or sequence of instructions on how the data is to be processed.

Configuration. The collection of brokers, their execu**tion groups, the message flows and sets that are assigned to the**m, and the topics and associated access control specifications.

Connector. See Message processing node connector.

Cube. Another term for a fact table. It can represent “n” dimensions, rather than just three (as may be implied by the name).

DDL (Data Definition Language). a SQL statement that creates or modifies the structure of a table or database. For example, CREATE TABLE, DROP TABLE, ALTER TABLE, CREATE DATABASE.

DML (Data Manipulation Language). an INSERT, UPDATE, DELETE, or SELECT SQL statement.

Data Append. A data loading technique where new data is added to the database leaving the existing data unaltered.

Data Cleansing. A process of data manipulation and transformation to eliminate variations and inconsistencies in data content. This is typically to improve the quality, consistency, and usability of the data.

Data Federation. The process of enabling data from multiple heterogeneous data sources to appear as though it is contained in a single relational database. Can also be referred to “distributed access”.

Data mart. An implementation of a data warehouse, typically with a smaller and more tightly restricted scope - such as for a department, workgroup, or subject area. It could be independent, or derived from another data warehouse environment (dependent).

Data mart - Dependent. A data mart that is consistent with, and extracts its data from, a data warehouse.

Data mart - Independent. A data mart that is standalone, and does not conform with any other data mart or data warehouse.

Data Mining. A mode of data analysis that has a focus on the discovery of new information, such as unknown facts, data relationships, or data patterns.

Data Model. A representation of data, its definition, characteristics, and relationships.

Data Partition. A segment of a database that can be accessed and operated on independently even though it is part of a larger data structure.

Data Refresh. A data loading technique where all the data in a database is completely replaced with a new set of data.

Data silo. A standalone set of data in a particular department or organization used for analysis, but typically not shared with other departments or organizations in the enterprise.

Data Warehouse. A specialized data environment developed, structured, shared, and used specifically for decision support and informational (analytic) applications. It is subject oriented rather than application oriented, and is integrated, non-volatile, and time variant.

Database Instance. A specific independent implementation of a DBMS in a specific environment. For example, there might be an independent DB2 DBMS implementation on a Linux server in Boston supporting the Eastern offices, and another separate and independent DB2 DBMS on the same Linux server supporting the western offices. They would represent two instances of DB2.

Database Partition. Part of a database that consists of its own data, indexes, configuration files, and transaction logs.

DataBlades. These are program modules that provide extended capabilities for Informix databases, and are tightly integrated with the DBMS.

DB Connect. Enables connection to several relational database systems and the transfer of data from these database systems into the SAP Business Information Warehouse.

Debugger. A facility on the Message Flows view in the Control Center that enables message flows to be visually debugged.

Deploy. Make operational the configuration and topology of the broker domain.

Dimension. Data that further qualifies and/or describes a measure, such as amounts or durations.

Distributed Application. In message queuing, a set of application programs that can each be connected to a different queue manager, but that collectively constitute a single application.

Distribution List. A list of MQSeries® queues to which a message can be put using a single statement.

Drill-down. Iterative analysis, exploring facts at more detailed levels of the dimension hierarchies.

Dynamic SQL. SQL that is interpreted during execution of the statement.

Element. A unit of data within a message that has a business meaning.

Engine. A program that performs a core or essential function for other programs. A database engine performs database functions on behalf of the database user programs.

Enqueue. To put a message on a queue.

Enrichment. The creation of derived data. An attribute level transformation performed by some type of algorithm to create one or more new (derived) attributes.

Entity. A person, place, thing, or event of interest to the enterprise. Each entity in a data model is unique.

Event Queue. The queue onto which the queue manager puts an event message after it detects an event. Each category of event (queue manager, performance, or channel event) has its own event queue.

Extenders. These are program modules that provide extended capabilities for DB2, and are tightly integrated with DB2.

FACTS. A collection of measures, and the information to interpret those measures in a given context.

Federated data. A set of physically separate data structures that are logically linked together by some mechanism, for analysis, but which remain physically in place.

Federated Server. Any DB2 server where the WebSphere Information Integrator is installed.

Federation. Providing a unified interface to diverse data.

Foreign Key. An attribute or set of attributes that refer to the primary key of another entity.

Framework. In WebSphere MQ, a collection of programming interfaces that allow customers or vendors to write programs that extend or replace certain functions provided in WebSphere MQ products.

Gateway. A means to access a heterogeneous data source. It can use native access or ODBC technology.

Grain. The fundamental lowest level of data represented in a dimensional fact table.

Input node. A message flow node that represents a source of messages for the message flow.

Instance. A particular realization of a computer process. Relative to database, the realization of a complete database environment.

Java Database Connectivity. An application programming interface that has the same characteristics as ODBC but is specifically designed for use by Java database applications.

Java Development Kit. Software package used to write, compile, debug and run Java applets and applications.

Java Message Service. An application programming interface that provides Java language functions for handling messages.

Java Runtime Environment. A subset of the Java Development Kit that allows you to run Java applets and applications.

Key. An attribute of set of attributes that uniquely identifies an entity.

Listener. In WebSphere MQ distributed queuing, a program that detects incoming network requests and starts the associated channel.

Materialized Query Table. A table where the results of a query are stored, for later reuse.

Measure. A data item that measures the performance or behavior of business processes.

Message broker. A set of execution processes hosting one or more message flows.

Message domain. The value that determines how the message is interpreted (parsed).

Message flow. A directed graph that represents the set of activities performed on a message or event as it passes through a broker. A message flow consists of a set of message processing nodes and message processing connectors.

Message parser. A program that interprets the bit stream of an incoming message and creates an internal representation of the message in a tree structure. A parser is also responsible to generate a bit stream for an outgoing message from the internal representation.

Message processing node connector. An entity that interprets the bit stream of an incoming message and creates an internal representation of the message in a tree structure. A parser is also responsible to generate a bit stream for an outgoing message from the internal representation.

Message processing node. A node in the message flow, representing a well defined processing stage. A message processing node can be one of several primitive types or it can represent a subflow.

Message Queue Interface. The programming interface provided by the WebSphere MQ queue managers. This programming interface allows application programs to access message queuing services.

Message queuing. A communications technique that uses asynchronous messages for communication between software components.

Message repository. A database holding message template definitions.

Message set. A grouping of related messages.

Message type. The logical structure of the data within a message.

Metadata. Typically called data (or information) about data. It describes or defines data elements.

MOLAP. Multi-dimensional OLAP. Can be called MD-OLAP. It is OLAP that uses a multi-dimensional database as the underlying data structure.

MQSeries. A previous name for WebSphere MQ.

Multi-dimensional analysis. Analysis of data along several dimensions. For example, analyzing revenue by product, store, and date.

Multi-Tasking. Operating system capability which allows multiple tasks to run concurrently, taking turns using the resources of the computer.

Multi-Threading. Operating system capability that enables multiple concurrent users to use the same program. This saves the overhead of initiating the program multiple times.

Nickname. An identifier that is used to reference the object located at the data source that you want to access.

Node Group. Group of one or more database partitions.

Node. See Message processing node and Plug-in node.

Non-Additive Measure. Measure of a fact that cannot be added across any of its dimensions, such as a percentage.

ODS. (1) Operational data store: A relational table for holding clean data to load into InfoCubes, and can support some query activity. (2) Online Dynamic Server - an older name for IDS.

OLAP. OnLine Analytical Processing. Multi-dimensional data analysis, performed in real-time. Not dependent on underlying data schema.

Open Database Connectivity. A standard application programming interface for accessing data in both relational and non-relational database management systems. Using this API, database applications can access data stored in database management systems on a variety of computers even if each database management system uses a different data storage format and programming interface. ODBC is based on the call level interface (CLI) specification of the X/Open SQL Access Group.

Optimization. The capability to enable a process to execute and perform in such a way as to maximize performance, minimize resource utilization, and minimize the process execution response time delivered to the end user.

Partition. Part of a database that consists of its own data, indexes, configuration files, and transaction logs.

Pass-through. The act of passing the SQL for an operation directly to the data source without being changed by the federation server.

Pivoting. Analysis operation where user takes a different viewpoint of the results. For example, by changing the way the dimensions are arranged.

Plug-in node. An extension to the broker, written by a third-part developer, to provide a new message processing node or message parser in addition to those supplied with the product.

PMML. An XML-based language that enables the definition and sharing of predictive models between applications.

Point-to-point. Style of application messaging in which the sending application knows the destination of the message.

Predefined message. A message with a structure that is defined before the message is created or referenced.

Primary Key. Field in a table record that is uniquely different for each record in the table.

Process. An instance of a program running in a computer.

Program. A specific set of ordered operations for a computer to perform.

Pushdown. The act of optimizing a data operation by pushing the SQL down to the lowest point in the federated architecture where that operation can be executed. More simply, a pushdown operation is one that is executed at a remote server.

Queue Manager. A subsystem that provides queuing services to applications. It provides an application programming interface so that applications can access messages on the queues that are owned and managed by the queue manager.

Queue. A WebSphere MQ object. Applications can put messages on, and get messages from, a queue. A queue is owned and managed by a queue manager. A local queue is a type of queue that can contain a list of messages waiting to be processed. Other types of queues cannot contain messages but are used to point to other queues.

Relationship. The business rule that associates entities.

Right-time BI. BI that is available at the time it is required. That could be something between instantaneous and a long time. The objective is to understand the requirement for the specific BI and not assume that everything is required instantaneously. The focus should be to optimize the capabilities of the current technology, acceptable costs, and real business need.

ROLAP. Relational OLAP. Multi-dimensional analysis using a multi-dimensional view of relational data. A relational database is used as the underlying data structure.

Roll-up. Iterative analysis, exploring facts at a higher level of summarization.

Semi-additive Measure. Measure of a fact that can be added across only some of its dimensions, such as a balance.

Server. A computer program that provides services to other computer programs (and their users) in the same or other computers. However, the computer that a server program runs in is also frequently referred to as a server.

Shared nothing. A data management architecture where nothing is shared between processes. Each process has its own processor, memory, and disk space.

Slice and Dice. Analysis across several dimensions and across many categories of data items. Typically to uncover business behavior and rules.

Static SQL. SQL that has been compiled prior to execution. Typically provides best performance.

Static SQL. SQL that has been compiled prior to execution. Typically provides best performance.

Subflow. A sequence of message processing nodes that can be included within a message flow.

Subject Area. A logical grouping of data by categories, such as customers or items.

Synchronous Messaging. A method of communication between programs in which a program places a message on a message queue and then waits for a reply before resuming its own processing.

Task. The basic unit of programming that an operating system controls. Also see Multi-Tasking.

Thread. The placeholder information associated with a single use of a program that can handle multiple concurrent users. Also see Multi-Threading.

Type Mapping. The mapping of a specific data source type to a DB2 UDB data type

Unit of Work. A recoverable sequence of operations performed by an application between two points of consistency.

User Mapping. An association made between the federated server user ID and password and the data source (to be accessed) user ID and password.

Virtual Database. A federation of multiple heterogeneous relational databases.

Warehouse Catalog. A subsystem that stores and manages all the system metadata.

WebSphere MQ. A family of IBM licensed programs that provide message queuing services.

Wrapper. The means by which a data federation engine interacts with heterogeneous sources of data. Wrappers take the SQL that the federation engine uses and maps it to the API of the data source to be accessed. For example, they take DB2 SQL and transform it to the language understood by the data source to be accessed.

xtree. A query-tree tool that allows you to monitor the query plan execution of individual queries in a graphical environment.

Zero latency. This is a term applied to a process where there are no delays as it goes from start to completion.

Abbreviations and acronyms

ACS	Access Control System
ADK	Archive Development Kit
AIX	Advanced Interactive eXecutive from IBM
AMI	Application Messaging Interface
API	Application Programming Interface
AQR	Automatic Query Re-write
AR	Access Register
ARM	Automatic Restart Manager
ART	Access Register Translation
ASCII	American Standard Code for Information Interchange
AST	Application Summary Table
BI	Business Intelligence
BLOB	Binary Large OBject
BW	Business Information Warehouse (SAP)
CCMS	Computing Center Management System
CFG	Configuration
CLI	Call Level Interface
CLOB	Character Large OBject
CLP	Command Line Processor
CORBA	Common Object Request Broker Architecture
CPU	Central Processing Unit
CRM	Customer Relationship Management
CS	Cursor Stability
DAS	DB2 Administration Server
DB	Database
DB2	Database 2™
DB2 UDB	DB2 Universal DataBase
DBA	Database Administrator
DBM	DataBase Manager
DBMS	DataBase Management System
DCE	Distributed Computing Environment
DCM	Dynamic Coserver Management
DCOM	Distributed Component Object Model
DDL	Data Definition Language - a SQL statement that creates or modifies the structure of a table or database. For example, CREATE TABLE, DROP TABLE.
DES	Data Encryption Standard
DIMID	Dimension Identifier
DLL	Dynamically Linked Library
DML	Data Manipulation Language - an INSERT, UPDATE, DELETE, or SELECT SQL statement
DMS	Database Managed Space
DPF	Data Partitioning Facility
DRDA	Distributed Relational Database Architecture™
DSA	Dynamic Scalable Architecture
DSN	Data Source Name
DSS	Decision Support System
EAI	Enterprise Application Integration
EBCDIC	Extended Binary Coded Decimal Interchange Code

EDA	Enterprise Data Architecture
EDU	Engine Dispatchable Unit
EDW	Enterprise Data Warehouse
EGM	Enterprise Gateway Manager
EJB	Enterprise Java Beans
ER	Enterprise Replication
ERP	Enterprise Resource Planning
ESE	Enterprise Server Edition
ETL	Extract, Transform, and Load
ETTL	Extract, Transform/Transport, and Load
FP	Fix Pack
FTP	File Transfer Protocol
Gb	Gigabit
GB	Gigabyte
GUI	Graphical User Interface
HADR	High Availability Disaster Recovery
HDR	High availability Data Replication
HPL	High Performance Loader
I/O	Input/Output
IBM	International Business Machines Corporation
ID	Identifier
IDE	Integrated Development Environment
IDL	Interface Definition Language
IDS	Informix Dynamic Server
II	Information Integrator
IMG	Integrated Implementation Guide (for SAP)
IMS	Information Management System
ISAM	Indexed Sequential Access Method
ISM	Informix Storage Manager
ISV	Independent Software Vendor
IBM	International Business Machines Corporation
IT	Information Technology
ITR	Internal Throughput Rate
ITSO	International Technical Support Organization
IX	Index
J2EE	Java 2 Platform Enterprise Edition
JAR	Java Archive
JDBC	Java DataBase Connectivity
JDK™	Java Development Kit
JE	Java Edition
JMS	Java Message Service
JRE™	Java Runtime Environment
JVM™	Java Virtual Machine
KB	Kilobyte (1024 bytes)
LDAP	Lightweight Directory Access Protocol
LPAR	Logical Partition
LV	Logical Volume
Mb	Megabit
MB	Megabyte
MDC	Multidimensional Clustering
MPP	Massively Parallel Processing
MQI	Message Queuing Interface
MQT	Materialized Query Table
MRM	Message Repository Manager
MTK	DB2 Migration ToolKit for Informix
NPI	Non-Partitioning Index
ODBC	Open DataBase Connectivity
ODS	Operational Data Store
OLAP	OnLine Analytical Processing
OLE	Object Linking and Embedding

OLTP	OnLine Transaction Processing
ORDBMS	Object Relational DataBase Management System
OS	Operating System
PDS	Partitioned Data Set
PIB	Parallel Index Build
PMML	Predictive Model Markup Language
PSA	Persistent Staging Area
RBA	Relative Byte Address
RDBMS	Relational DataBase Management System
RID	Record Identifier
RR	Repeatable Read
RS	Read Stability
SCB	Session Control Block
SDK	Software Developers Kit
SID	Surrogage Identifier
SMIT	Systems Management Interface Tool
SMP	Symmetric MultiProcessing
SMS	System Managed Space
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SPL	Stored Procedure Language
SQL	Structured Query
TCB	Thread Control Block
TMU	Table Management Utility
TS	Tablespace
UDB	Universal DataBase
UDF	User Defined Function
UDR	User Defined Routine
URL	Uniform Resource Locator
VG	Volume Group (Raid disk terminology)
VLDB	Very Large DataBase
VP	Virtual Processor
VSAM	Virtual Sequential Access Method
VTI	Virtual Table Interface
WSDL	Web Services Definition Language
WWW	World Wide Web
XBSA	X-Open Backup and Restore APIs
XML	eXtensible Markup Language
XPS	Informix eXtended Parallel Server

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information on ordering these publications, see "How to get IBM Redbooks" on page 362. Note that some of the documents referenced here may be available in softcopy only.

- *Business Performance Management...Meets Business Intelligence*, SG24-6340
- *Preparing for DB2 Near-Realtime Business Intelligence*, SG24-6071
- *Data Mart Consolidation: Getting Control of Your Enterprise Information*, SG24-6653
- *WebSphere Information Integrator Q Replication: Fast Track Implementation Scenarios*, SG24-6487
- *Using Web Services for Business Integration*, SG24-6583
- *A Practical Guide to DB2 UDB Data Replication V8*, SG24-6828
- *Building the Operational Data Store on DB2 UDB Using IBM Data Replication, WebSphere MQ Family, and DB2 Warehouse Manager*, SG24-6513

Online resources

These Web sites and URLs are also relevant as further information sources:

- Use DB2 Alphablox to generate reports for your business processes.

  `http://www-128.ibm.com/developerworks/ibm/library/i-odoebp12`

- Right-time Information, right now. How to turn the data you get into the information you need.

  `http://www-306.ibm.com/e-business/ondemand/us/governance/righttime.shtml/`

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

A

abstraction layer 15
Action Manager 198, 200–201
Adaptive Action Manager 198, 200–201, 203
aggregation functions 122
AIX 120, 236, 270, 316
alerts 23, 25–26, 97, 101, 151, 187, 265, 268, 277, 350
Alerts view 201
Alphablox 118–120, 148–160, 164, 166, 198, 201, 203–204, 208, 265, 267–268, 270–271, 319–321, 324, 346
Alphablox architecture 149
AMI 242–243
analytic application 7, 9, 23, 25–26, 42, 50, 68, 75, 99–100, 102, 108, 157, 159, 166, 248, 268, 270, 278, 280
analytic structures 119
analytical data 254
analytical functions 79
analytical service 100
API 20, 121–122, 140, 162, 164, 173, 175, 181, 236, 258, 288–289, 324
application development 77, 191
application logic 106, 165, 298
Application Messaging Interface 242
Apply function 241
architecture ix, 5, 10, 16, 27, 38, 44–45, 63–64, 73–74, 76, 78–86, 88, 90–93, 97, 104–105, 107–108, 118–119, 155, 158, 164, 177, 179, 196–197, 200–201, 208, 216, 218, 244, 250, 265–266, 270–271, 303, 305, 307, 313, 318
artifact modeling 189
artifacts 190, 199, 211
awk 139

B

Basel II 10
batch environment 38
best practices 14, 62, 78, 187, 218
BI ix, 2–5, 24–25, 46, 50, 57, 59, 61, 65, 82, 98, 100, 112–113, 124, 126, 134, 217–219, 270, 276, 284, 286
BIO 188
BMC 20
BPEL 89, 189–193, 201, 208, 210–213
BPM 2–3, 47–48, 98
BPU 219–220
brokers 225, 229, 251
building blox 119, 158
business analysis 189
business integration 28, 174, 178, 187, 189, 191, 194, 209, 211, 217
  connect 187
business intelligence ix, 2–5, 8, 13, 22–23, 25, 27, 39, 43, 50, 54, 57, 61, 65, 72, 75, 79, 81, 84, 86, 97, 99–100, 111–112, 114–115, 117, 121, 124, 168–169, 176, 182, 201, 218, 223–224, 231, 252–254, 265–266, 281, 285, 287, 349
  also see BI
  operational 25, 223
  real-time 43, 79, 81, 223
  strategic 25
  tactical 25
business measurements and goals 5, 281
business measures 187–188, 191, 195–200, 202–206
business measures model 191, 195–196, 199, 202, 205
  monitor 191, 199
business modeling 187, 194
business operations 102, 104, 195
business performance 3–4, 25, 48, 62, 77, 195, 204, 207, 248
business process ix, 3–4, 6–9, 15, 18–23, 29–30, 36, 41, 47–49, 57, 77, 95–98, 100, 105, 165, 167, 169, 175, 178, 186–190, 193–196, 199, 208, 210–214, 238, 271, 273, 275, 324
Business Process Execution Language 189–190, 208
business process management 97
business process reuse 104
business rules 23, 96–97, 185, 209, 212, 273
business strategy 186
business transformation 168
business value chain 285

C
CA-Datacom 291
CA-IDMS 291
case study 1, 28, 109, 261–263, 266, 274, 281–285, 287, 291, 295, 297–298, 300–301, 305, 313, 319–320, 341, 349
CBE 198, 200
CDTA 245
CEI 198, 200
changed data capture 246, 343
CLI 117, 121–122
closed-loop 3, 25, 34, 43, 50, 60, 92, 102, 108, 265, 268, 271, 281, 285, 287, 290, 292, 295, 312, 350
closed-loop process 50, 53, 76, 103, 284, 350
Cloudscape 21
CLP 139
clustering 115, 137, 209, 236
clustering index 94
collocated tables 137
collocation 137
common base event 200
common business event 198
common data elements 171
common event infrastructure 198, 200
component reuse 105
concurrency 94, 241, 253
concurrent 7, 46, 162, 221, 245
containers 128, 156–157, 256
continuous update 39, 251, 254
control flow 131–132, 145–147
conversion 45, 172
coordinator partition 136
corporate portals 168
cube modeling 118
Cube Views 112–115, 118, 142, 197–198, 201, 204, 208
customer relationship management 32

D
dashboards 3, 7, 26, 43, 46, 82, 84, 87, 101, 119, 150, 198, 201–204, 268
data architecture 44, 81
data availability 92, 229
data capture 90, 92, 176, 231, 246, 343
data cleansing 171–172, 230
data consistency 87, 176–177
data consolidation 77
Data Definition Language 184
data definitions 74
data elements 87, 171, 255
data federation 40, 46, 286–288, 340, 350
data flows 91, 119, 130–131, 140, 145–147, 253–254
data integration 3, 70, 86–87, 96, 174, 179–181, 249, 254
data integrity 97, 225, 227
data latency 88
data mart 113, 257
data mart consolidation 41
data mart proliferation 41
data mining 7, 23, 44, 60, 71, 79, 82, 100, 108, 112–113, 115, 117–119, 126, 132, 140, 270, 311, 349
data model 23, 38, 44, 54–56, 77, 80, 84, 115, 183–184
data partitioning 112
data quality 79, 96, 180, 340–341
data redundancy 36, 41, 98
data replication 90–92, 121, 286, 300, 305, 313, 343, 350
data silos 16, 74–75, 77
data transformation 38, 130, 146, 172, 179, 185, 214, 230, 233, 249, 269, 344
data types 172, 182, 255, 340
data warehouse 9, 13, 23–26, 28, 34, 37–40, 42, 45–46, 48, 55, 57, 59, 61–62, 64, 66, 68, 75–76, 79, 81–84, 87, 91, 93–94, 97–100, 102, 107–108, 113, 117, 119, 121–122, 132, 145, 179, 218, 221, 223–224, 240–241, 244–245, 247–250, 252–255, 257, 265–266, 268–270, 272, 276, 279, 316, 340, 342–346, 349
  architecture 38
  strategic decision-making 38
data warehousing 3, 7, 24–25, 27, 29, 38, 40, 55, 66–67, 72, 75–77, 79, 81, 91–94, 99, 103, 112–113, 117–118, 132, 217–219, 223–224, 244–245, 247, 250, 275–276, 300
database
  History 203
  Monitor 201–204
  Repository 160, 202
  Runtime 201–203
  State 202–203
database partition 135–137, 220
database partitioning 67, 94
database schema 182, 202, 204
DataStage 144, 147, 169, 172, 179–181, 184, 224,

252, 254–259, 267, 270, 291, 314–316, 340–341, 343–345
DB2 Alphablox 148, 150, 152–159, 198, 201, 203, 220, 270–271, 320
DB2 Alphablox analytic-enabled solutions 165
DB2 Alphablox application deployment 160
DB2 Alphablox applications 151
DB2 Alphablox architecture 149
DB2 Alphablox data sources 164
DB2 Alphablox repository 158, 160, 163, 167
DB2 Alphablox security 166
DB2 Alphablox server 163
DB2 Alphablox services 161–162
DB2 catalog 122, 220
DB2 Connect 121, 267
DB2 Content Manager 175
DB2 Cube Views 113–114, 198, 201
DB2 Data Warehouse Center 117
DB2 Data Warehouse Edition 62, 111–112, 117, 119
also see DWE
DB2 Data Warehouse Enterprise Edition 120
DB2 ESE 120
DB2 for z/OS 290
DB2 Intelligent Miner 113, 115, 118, 140
DB2 MQ functions 241–244
DB2 MQ Listener 239
Stored Procedure 239
DB2 Office Connect Enterprise Web Edition 112
DB2 OLAP Server 112–113
DB2 optimizer 118
DB2 Query Patroller 112, 116, 120, 137–138
DB2 replication services 201
DB2 staging table 146
DB2 UDB 112–117, 120–122, 145, 178–179, 208, 224, 229–234, 242–243, 262–263, 267, 270–271, 274, 289, 306–307, 313, 315, 339
DB2 UDB Database Partitioning Feature 112–113
DB2 UDB Enterprise Server Edition 112–113
DB2 Universal Database 112, 121, 174, 177, 180, 209, 313
DB2 Warehouse Manager 112, 116
DDL 126, 184
decision making 3, 38, 48, 53, 61, 74–76, 84, 100, 150, 178, 268, 270, 346
deploy 145
Derby 21
dimensions 62, 67–68, 101, 118, 164, 168, 207
discrete business process reuse 104
disparate systems 181–182
DPF 136–137, 220
DRDA wrapper 289
drill-down 206
drop 47, 151, 181, 184, 210, 257
DWE 62, 112–114, 117–124, 130–133, 139–146, 218, 270
DWE Admin Console 144
DWE Design Studio 119–120, 122–123, 144, 146
DWE Integrated Installer 120, 133
DWE Mining 120, 140
DWE OLAP 118, 121–122
DWE SQL Warehousing Tool 120, 144
DWL 22
Dynamic HTML 151, 160

E

EAI 53, 80, 87–89, 91–93, 107, 214, 247, 251, 287, 324
also see Enterprise Application Integration
Easy Mining Procedures 140
Eclipse 20, 118–119, 123, 144, 187, 189, 191, 208, 214–215
EDW 7, 29, 40, 46, 76, 284, 286–287
also see enterprise data warehouse
EII 80, 85–87, 91–93, 107–108, 291, 340, 342
also see Enterprise Information Integration
ELT 7, 24, 38, 46, 85, 224, 245, 252
also see extract, load, and transform
embedded SQL 121–122
Enterprise Application Integration 24, 80, 88, 247
also see EAI
enterprise data warehouse 7, 41, 46, 76, 83, 268, 270
also see EDW
Enterprise Information Integration 44, 78, 80, 172, 288
also see EII
enterprise information integration
also see EII
Enterprise Information Systems 156, 213
Enterprise Java Beans 143, 163
enterprise service bus 38, 41–42, 106–107, 208, 214, 216, 261, 269, 271–272, 287, 339
ETL 7, 23–24, 38–39, 46, 76–77, 79–80, 84–85, 87–88, 91–93, 97–98, 107–108, 112–113, 117, 144, 172, 174, 178, 184, 220, 224, 250, 252–254, 270, 284, 286, 291, 313–316, 340–342, 350

also see extract, transform, and load
ETL processes 251
ETL tool 92, 224, 253
Event 23
event driven 23, 26
events 15, 23, 25, 37, 47, 54, 75, 77, 92, 158, 175, 178, 191, 195–200, 202, 214, 225, 258, 269, 283, 309, 312–313, 350
Excel wrapper 296
Extensible Stylesheet Language 172
extract, load, and transform 7, 24, 252
also see ELT
extract, transform, and load 38, 76, 79–80, 112, 117, 224, 254, 257
also see ETL

F

facts 62, 95–96
federated access 173, 249
federated data access 87, 112
federated server 288–289, 293, 295
federation 3, 40, 46, 86, 111, 169, 172–175, 178, 212, 247, 286–288, 340–341, 349–350
FileNet 20
flat files 169, 174, 259
Flow Definition Language 189
functional automation 32

G

GUI 123, 144, 176, 181, 185, 254
guided analysis 62, 101–102, 151

H

hardware cost 23
hashing algorithm 137
high availability 113, 218, 237
History database 203–204
HP-UX 120, 236
HTTP Server 141
human task 211

I

IBM Content Manager 10
IBM IMS 291, 340
IBM Informix 174, 180
IBM Tivoli 20
IBM WebSphere DataStage 172, 179
IBM WebSphere Information Integrator 117, 169, 179
IBM WebSphere Information Integrator Classic Federation for z/OS 175, 179
IBM WebSphere Information Integrator Content Edition 175
IBM WebSphere Information Integrator Standard and Advanced Editions 175
IBM WebSphere ProfileStage 171
IBM WebSphere QualityStage 172
IBM z/OS 169
import 192–193
importing 190, 212
inconsistent data 98, 171
independent data marts 41
indexes 94, 116, 135–136, 138
information as a service ix, 8, 10–11, 14, 17, 20, 100, 103
information integration 85, 111, 167–170, 172, 175, 179, 288, 340–341
information management 6, 15, 58, 79, 168
information on demand 10, 17, 19–20, 23, 79, 168
information pyramid 44, 82
information silos 16
information technology 3, 29
also see IT
Informix 21
infrastructure rationalization 168
in-line 8–9, 23, 100, 112, 172
in-line analytics 100
innovation 30, 32, 188
integrated information infrastructure 8
IT ix, 3, 7, 17, 20, 23, 32, 34–36, 38, 47, 76–77, 79, 88, 90, 97, 103–104, 106, 169, 171, 174, 176, 187–189, 208, 210, 214–217, 270, 317, 350

J

J2EE 119, 142–143, 148–150, 155–156, 158–159, 162, 166, 178, 209, 213, 216, 238
Java 115, 140–143, 156–157, 159–160, 174, 178, 209, 211, 214, 217, 236, 306, 337, 339
Java Server Faces 142, 211
Java visualizer 140
JDBC 121–122, 174, 289, 291
Just-in-time 31

K

Kanban 31

key performance indicators 23, 25, 42–43, 47, 80, 150, 152, 172, 187, 195, 198, 202, 268, 277
also see KPI
KPI 47, 150, 191, 197, 205, 277
also see key performance indicators

L
latency ix, 24, 29–30, 48, 54, 68, 75, 79, 88, 92, 95, 102, 107–108, 176, 224, 226, 235, 245, 247–249, 284, 286–287, 289–291, 298, 300, 305, 309, 312, 350
launchpad installation tool 207
layered data architecture 44
LDAP 163
Lightweight Directory Access Protocol 117, 162
Linux 120, 169, 179, 230, 232, 236, 262–263, 289, 315
load 84–85, 92, 95, 107, 146, 177, 185, 251, 314
LOB data type 136
Lotus Workplace 318

M
maintenance costs 41, 106, 179
Master Data Management 168
materialized query tables 118
MDC 94
also see multidimensional clustering
MDM 86, 95–98
also see master data management
metadata 15, 18, 22, 74, 84, 87, 92, 97–98, 114, 119, 121–122, 160, 165, 171, 175, 182, 197, 202, 244
metadata objects 121–122
metadata repository 92
metric 189, 191, 200–201
Microsoft 114, 117, 120, 169, 174, 177, 230, 232, 236, 264, 289, 295–296
Excel 117, 289
SQL Server 295
Microsoft SQL Server 180, 264, 295
Microsoft Windows 169
modeling 56, 112–113, 115, 127–128, 139, 190, 193
MPP 256
MQ Series 242
MQT 118
multidimensional 94
multidimensional analysis 62, 79, 151, 164
multidimensional clustering 93–94
also see MDC
multidimensional data 121, 152, 164
multidimensional data servers 156
multidimensional databases 150
multidimensional metadata 121
multidimensional reports 203
multidimensional view 207

N
near real-time 2, 38–39, 223, 225, 244, 248–249, 251, 254
nickname 290, 294–295, 298
normalization 183–184

O
ODBC 117, 121–122, 174, 256, 258, 289, 291–292, 316
ODS 7, 24–25, 29, 40, 46, 75, 83–85, 91, 223, 253–254, 268, 276, 279, 284, 286–287, 301, 309–311, 314, 316–317, 328, 339–340, 349
also see operational data store
OLAP 23, 44, 79, 82, 101, 112–114, 117–123, 129, 132, 142, 144, 270
OLAP Center 121–122
OLAP objects 122
OLTP 113, 115, 300
on demand 27
on demand enterprise 6, 120, 214
on demand real-time enterprise 1–2, 6–7, 25, 37, 64, 106, 265
operational data 69, 71, 83, 340–342, 346
operational data store 7, 24–25, 29, 46, 75, 83–84, 176, 178, 223–224, 235, 268, 276, 314, 316, 342, 345
also see ODS
operational systems 24, 38, 54, 76, 80–82, 84, 245–247, 275–277
operational transaction systems 9
Optimization Advisor 114, 121–122
Oracle 20, 117, 169, 174, 177, 180, 184, 230, 232, 263, 289, 292–295
organization modeling 189

P
parallelism 177
partitioned database 135–136, 251

partitioned storage model 136
partitioning key 136–137
PeopleSoft 174, 180
performance warehouse 207
Perl 139
physical data modeling 118
platform 237
portal 96, 148, 169, 194, 201, 204, 207–208, 265–266, 271, 317–321, 346
PortletLink 324
Predictive Model Markup Language 140
prescriptive analytics 18
PresentBlox 322
proactive approach 42
process engine 177, 210
process management 4, 49–50, 97–98, 156, 175
process monitoring 46
ProfileStage 169, 171, 181–184
publishing server 194

Q

Query Patroller Center 138–139
Query Patroller command line support 139
Query Patroller server 138

R

Rational Application Developer 210
Rational ClearCase 191
Rational Data Architect 118
Rational Software Architect 190
Rational XDE 190–191
real-time ix, 1–2, 4–8, 10, 22–30, 34, 36–43, 45–48, 52–54, 56–61, 64–65, 68, 70, 72–76, 78–79, 81, 84, 87–92, 96–97, 100, 102–103, 106, 108–109, 111, 115, 121, 150–152, 155, 169, 172, 176, 179–180, 185, 189, 215, 217, 223–224, 244–245, 247–250, 254, 261, 265–266, 269, 273–274, 278–282, 284–287, 291, 297, 317, 335, 340–341, 343, 345, 349–350
real-time analytics 23, 100
real-time business intelligence 27
real-time data 29, 37, 40, 43, 46–47, 223–224, 244–245, 249, 251, 281, 284, 340
real-time data warehouse 179
real-time data warehousing 245
real-time enterprise x, 1–2, 4–8, 23, 25, 27–30, 36–37, 40, 43, 46, 48, 54, 59–60, 64–65, 68, 73–75, 78, 84, 88, 100, 103, 108–109, 217, 223–224, 261, 265–266, 274, 279–282, 284, 350
  also see RTE
real-time environment 26, 59
real-time views 175
Redbooks Web site 362
  Contact us xiii
redundant data 77, 83, 183
replication 86, 90–92, 121, 146, 169, 172, 176–177, 197, 201, 224, 226–228, 231–232, 234–235, 251, 286, 298–305, 308, 313, 343, 350
  queue 227, 232
  SQL 177, 224, 298–299
replication setup 301, 305
reporting environment 296
reporting tools 148, 174
reports 43, 75, 84, 138–139, 150, 174, 182, 184, 190, 192, 195, 203, 207, 211, 268, 322–324, 327, 332, 340, 346, 349
Repository database 202
resource modeling 189
reverse engineering 127, 132
REXX 139
right-time data 63, 68–70, 72, 244
risk and compliance 168
ROI 19, 190
RTE x
  also see real-time enterprise
RTI 340
  also see WebSphere Real Time Integration
rules-based 23, 103, 178
Runtime database 202–204

S

sales force automation 32
SAP 169, 174, 178, 180, 307
Sarbanes-Oxley 191
SAS 180
scalability 88, 113, 218, 236, 238
scalable vector graphics 194
schedules 116
schema generator 197, 204
schemas 23, 203, 336–337
scorecard 47–48, 204
scoring 112–113, 115, 139, 141–142
service component architecture 190
service integration bus 200
service management 20
service oriented architecture ix, 8, 17, 19, 103, 106,

214, 273
also see SOA
services
event 85, 201, 309
infrastructure 9, 34, 104, 174
partner 209
virtualization 20, 34
severity messages 127
Siebel 174, 180
single version of the truth 15
SMP 256
SOA ix, 8, 38, 91, 98, 103–107, 169, 174, 208–209, 214, 216, 272, 339, 343
also see service oriented architecture
reference architecture 104
Software AG Adabas 291
Solaris 120, 236
spreadsheets 96, 116, 169, 174, 296
SQL replication 177, 224, 226, 231, 234–235
SQL Server 2000 264
SQL Stored Procedures 140
SQL Warehousing 118–120, 123, 130, 142, 144
SQL Warehousing Tool 130, 144
also see SQW
SQW 130, 144–145, 148
also see SQL Warehousing Tool
SRD 22
staging area 24, 253
staging tables 39, 225, 229, 233–234
standardization 8, 30, 97, 170, 340
State database 202–203
Static HTML 160
statistics 117, 119, 140, 284, 286, 346, 348
stored procedures 117, 121–122, 138–141, 177, 226, 230, 232–233, 240–242, 244, 251
structured data 7, 83, 90, 172
summary tables 121–122
surrogate keys 250, 344
SVG Viewer 194
Sybase 117, 174, 177, 180, 230, 232
system-driven analytics 100

T

tablespaces 128
teradata 117, 174, 178, 180, 230, 232
text mining 17
total cost of ownership 77, 218, 270
training 193
transformation rules 92, 341
triggers 102, 187, 191, 200, 233, 246, 309, 312–313, 328, 340
Trigo 22

U

UDDI 297
UNIX 117, 120, 169, 179, 230, 232, 256
unstructured content 340
unstructured data 44, 77, 86, 90, 92, 117, 248
user-based complex analytics 101
user-based real-time analytics 100
user-defined functions 117, 139, 313

V

value chain 65–67, 285
Velocity 52
views 112–115, 118, 142, 197–198, 201, 204, 208
virtualization 21, 86–87, 339
visualization 112–113, 115, 139, 141
VSAM 291

W

Web service interface 298
Web services 3, 7–8, 10, 53, 85, 89–90, 92, 106, 172, 174, 180, 209, 213–214, 238, 247–248, 262–263, 275, 297, 313, 326–327, 333, 335–336, 339–340, 350
wrapper 297
Web Services Description Language 190, 297
WebSphere Application Server 118–119, 144–146, 148, 178, 200–201, 209–210, 213, 215–216, 220, 263, 271, 318, 336
WebSphere Business Integration 191
WebSphere Business Integration Adapters 213
WebSphere Business Integration Server 209
Foundation 209
WebSphere Business Integration Server Foundation 187
WebSphere Business Modeler 187–196, 200, 210
WebSphere Business Modeler Advanced 188–192, 196
simulation 189, 192
WebSphere Business Modeler Basic 188, 192
WebSphere Business Modeler Publishing Edition 188
WebSphere Business Modeler Publishing Server

188, 194
WebSphere Business Monitor 187, 191, 195–196, 198–200, 204, 207
  architecture 196
  components 196
  databases 197
  functionality 195, 200
WebSphere Business Monitor configurable views
  Active Instance 206
  Alert 207
  Dimensional 207
  Export actual values 207
  Gauge 205
  KPI 205
  Organization 207
  Process 207
  Report 207
  Scorecard 204
WebSphere Business Monitor server 200
WebSphere DataStage 179–180, 184, 255, 340
WebSphere DataStage Designer 181
WebSphere Enterprise Service Bus 111, 214, 216–217
WebSphere II 288, 307
  also see WebSphere Information Integration
WebSphere Information Integration 24, 168–169, 171–172, 176, 288
  also see WebSphere II
WebSphere Information Integration products 179
WebSphere Information Integrator 112, 117, 168–169, 172–173, 175, 177–179, 224, 262, 267, 270, 345–346
  configurable Web client 175
  content sources 174
  content-centric API 175
  event publishing 178
  federation 172–175
  life sciences sources 174
  mainframe databases 174
  packaged applications 174
  queue-based replication 177
  relational databases 174
  replication 176
  SQL-based replication 177
  transformation services 172
  wrappers 117
WebSphere Integration Developer 190, 208–210, 212–213
WebSphere Message Broker 41, 111, 214–217, 262, 267, 269–270, 274, 287, 307, 333, 338–339
WebSphere MQ 111, 174, 177–178, 180, 189, 191, 213–217, 224–225, 227, 229, 236–244, 262, 265, 267, 269–270, 274, 306–308, 333
WebSphere MQ Workflow 175, 191
WebSphere Portal 194, 204, 317–318
WebSphere Process Server 189, 191, 196, 199–200, 208–210, 212–213
  business processes 210
  business state machine 212
  data maps 212
  human tasks 211
  interface maps 212
  relationships 212
  selectors 213
  supporting components 212
  Web client 212
WebSphere ProfileStage 181–182, 184
WebSphere Real Time Integration 340
WebSphere Server Foundation 191
WebSphere Studio 190, 209
WebSphere Studio Application Developer 209
Windows 117, 120, 169, 179, 230, 232, 236, 263–264, 269, 271, 274, 289, 294
Windows 2000 120, 264
Windows NT 120
Windows Server 2003 120
Workflow automation 191
wrapper 117, 288–292, 295–298, 340
WS-BPEL 210, 212

X

XML 10, 90, 107, 117, 121–122, 159–160, 172, 174, 178, 180, 184, 190–191, 193, 200, 207, 225, 227, 238, 273, 289, 297–298, 307–309, 313, 320, 328–330, 333, 340, 342

Moving Forward with the On Demand Real-time Enterprise

(0.5” spine)
0.475”<->0.875”
250 <-> 459 pages

Moving Forward with the On Demand Real-time Enterprise

IBM Redbooks

Architecting the data warehousing environment for real-time

Modifying business processes to support continuous flow

Delivering business intelligence, on demand

Having more, and more current, information is fast becoming a requirement for business survival. The need permeates the enterprise because it enables proactive decision-making for problem avoidance, rather than reactive problem impact minimization. And that is the key. Thus the need for, and focus on, is real-time.

Companies no longer have the long strategic time-frames in which to plan, design, and manage their business processes. Yearly revenue goals and measurements are fast becoming quarterly goals and measurements. Investors and share-holders are more demanding. They are more critical, and less forgiving, of missed performance goals. And these demands are coming at a time when the volume of data is growing, there is an increase in business mergers and acquisitions, the use of strategic outsourcing is growing, and there is an increasing requirement for faster and faster turnaround on information requests. This has put an enormous burden on the information technology (IT) organizations. And most of this change is centered around business intelligence, because that is the environment responsible for providing information for management decision-making.

This IBM Redbook explores the techniques and capabilities for evolving to a real-time enterprise. It also demonstrates approaches for that evolution and provides examples to help guide you in developing your strategy and implementation methodology to become a real-time enterprise.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-7104-00 ISBN 0738494917